

Politechnika Warszawska  
Wydział Elektroniki i Technik Informacyjnych

---

# ANALITYCZNE BAZY DANYCH

## PROJEKT CZĘŚĆ 1

---

### ŹRÓDŁA DANYCH

## Spis treści

<b>1. Cel projektu</b>	2
<b>2. Baza relacyjna - Park Rozrywki</b>	2
2.1. Definicja systemu	2
2.2. Encje, związki oraz atrybuty	2
2.3. Klucze kandydujące i główne	10
2.4. Postać normalna	10
Postać Normalna - Kryterium pierwszej postaci normalnej:	10
Postać Normalna - Kryterium drugiej postaci normalnej:	10
Postać Normalna - Kryterium trzeciej postaci normalnej:	10
Więzy integralności:	11
2.5. Schemat modelu fizycznego	11
2.6. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	12
2.7. Skrypt w SQL	15
<b>3. Baza relacyjna - Akcje Marketingowe</b>	24
3.1. Definicja systemu	24
3.2. Encje, związki oraz atrybuty	24
3.3. Klucze kandydujące i główne	27
3.4. Postać normalna	27
Postać Normalna - Kryterium pierwszej postaci normalnej:	27
Postać Normalna - Kryterium drugiej postaci normalnej:	27
Postać Normalna - Kryterium trzeciej postaci normalnej:	27
Więzy integralności:	27
3.5. Schemat modelu fizycznego	28
3.6. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych	29
3.7. Skrypty w SQL	31

## 1. Cel projektu

Celem projektu było opracowanie oraz zaimplementowanie relacyjnej bazy danych przy użyciu narzędzi takich jak:

- Toad Data Modeler 7.2 do modelowania bazy danych,
- SQL Server Management Studio 19

Stworzona baza danych obejmuje obszar obsługi i kierowania firmą z sektora rozrywki, konkretnie – parkiem rozrywki. Dodatkowo istnieje druga baza danych skupiająca się na działaniach marketingowych realizowanych przez ten park.

## 2. Baza relacyjna - Park Rozrywki

### 2.1. Definicja systemu

W naszym systemie wyróżniamy następujące funkcjonalności:

- podgląd informacji o parkach rozrywki
- dodawanie/modyfikowanie/usuwanie informacji o parkach rozrywki
- podgląd informacji o właścicielach
- dodawanie/modyfikowanie/usuwanie informacji o właścicielach
- podgląd informacji o pracownikach
- dodawanie/modyfikowanie/usuwanie informacji o pracownikach
- podgląd informacji o wynagrodzeniach
- dodawanie/modyfikowanie/usuwanie informacji o wynagrodzeniach
- podgląd informacji o stanowiskach
- dodawanie/modyfikowanie/usuwanie informacji o stanowiskach
- podgląd informacji o punktach sprzedaży
- dodawanie/modyfikowanie/usuwanie informacji o punktach sprzedaży
- podgląd informacji o atrakcjach
- dodawanie/modyfikowanie/usuwanie informacji o atrakcjach
- podgląd informacji o biletach
- dodawanie/modyfikowanie/usuwanie informacji o biletach
- podgląd informacji o usługach
- dodawanie/modyfikowanie/usuwanie informacji o usługach
- podgląd informacji o klientach
- dodawanie/modyfikowanie/usuwanie informacji o klientach
- podgląd informacji o szkołach
- dodawanie/modyfikowanie/usuwanie informacji o szkołach
- podgląd informacji o osobach prywatnych
- dodawanie/modyfikowanie/usuwanie informacji o osobach prywatnych
- podgląd informacji o adresach
- dodawanie/modyfikowanie/usuwanie informacji o adresach
- podgląd informacji o pocztach
- dodawanie/modyfikowanie/usuwanie informacji o pocztach

### 2.2. Encje, związki oraz atrybuty

Definicja zbiorów encji określonych w projekcie:

- **ParkiRozrywki**, główna encja opisująca (domyślnie jeden) park rozrywki właściciela systemu.
- **Wlasciciele**, encja opisująca właścicieli parku rozrywki
- **Pracownicy**, encja opisująca pracowników danego parku rozrywki
- **Wynagrodzenia**, encja zawierająca dane o wynagrodzeniach pracowników
- **Stanowiska**, encja opisująca poszczególne stanowiska pracowników parku
- **PunktySprzedazy**, encja opisująca fizyczne punkty sprzedaży w parku rozrywki

- **Atrakcje**, encja zawierająca informacje o atrakcjach na terenie parku rozrywki
- **Bilety**, encja zawierająca informacje ogólne o sprzedanych biletach
- **Usługi**, encja zawierająca dane ogólne o usługach w parku rozrywki
- **Klienci**, encja opisująca klientów parku rozrywki
- **Szkoły**, encja uszczegóławiająca klientów-szkoły
- **OsobyPrywatne**, encja uszczegóławiająca klientów-osoby prywatne
- **Adresy**, encja zawierająca dane adresowe
- **Poczty**, encja uszczegóławiająca dane adresowe danymi poczty

Relacja	Nazwa	Krotność	Typ uczestnictwa
ParkiRozrywki – Pracownicy	Zatrudnia	jeden do wielu	park rozrywki musi istnieć, żeby mieć pracowników; pracownik nie musi istnieć, żeby park rozrywki istniał
ParkiRozrywki – Usługi	Oferuje	jeden do wielu	park rozrywki musi istnieć, żeby istniała usługa; usługa nie musi istnieć, żeby istniał park rozrywki
ParkiRozrywki – PunktySprzedazy	Posiada	jeden do wielu	park rozrywki musi istnieć, żeby istniał punkt sprzedaży; punkt sprzedaży nie musi występować w parku rozrywki
ParkiRozrywki – Klienci	Posiadanie Klientow	wiele do wielu	park rozrywki może mieć wielu klientów; jedna osoba może być klientem w więcej niż jednym parku rozrywki
ParkiRozrywki – Adresy	ParkRozrywki PosiadaAdres	jeden do jeden	każdy park rozrywki posiada dokładnie jeden adres
ParkiRozrywki – Wlasciciele	MaWlasciciela	jeden do wielu	park rozrywki może posiadać wielu właścicieli; jeden właściciel jest przypisany do dokładnie jednego parku rozrywki
ParkiRozrywki – Atrakcje	PosiadaAtrakcje	jeden do wielu	park rozrywki może posiadać wiele lub 0 atrakcji; jedna atrakcja należy tylko do jednego parku rozrywki

Tabela 1: Opis relacji między encjami w kontekście parku rozrywki (cz1).

<b>Relacja</b>	<b>Nazwa</b>	<b>Krotność</b>	<b>Typ uczestnictwa</b>
Pracownicy – Wynagrodzenia	PracownikMaWynagrodzenie	jeden do wielu	pracownik może posiadać 0 lub wiele wynagrodzeń; jedno wynagrodzenie należy do jednego pracownika
Pracownicy – Punkty Sprzedaży	PracujeW	jeden do wielu	każdy pracownik może (ale nie musi) pracować w dokładnie jednym punkcie sprzedaży; punktów sprzedaży może być 0 lub więcej
Pracownicy – Stanowiska	PracownikMaStanowisko	jeden do wielu	każdy pracownik może (ale nie musi) pracować w dokładnie jednym punkcie sprzedaży; stanowisk może być 0 lub więcej
Pracownicy – Adresy	PracownikPosiadaAdres	jeden do jeden	każdy pracownik posiada dokładnie jeden adres
Pracownicy – Usługi	RealizacjaUslugi	wiele do wielu	jeden pracownik może realizować jedną lub więcej usług; jedna usługa może być realizowana przez jednego lub więcej pracowników
Pracownicy – Atrakcje	ObslugaAtrakcji	wiele do wielu	jeden pracownik może obsługiwać jedną lub wiele atrakcji; jedną atrakcję może obsługiwać jeden lub więcej pracowników
Klienci – Usługi	WykorzystanieUslugi	wiele do wielu	jeden klient może korzystać z 0 lub więcej usług; z jednej usługi może korzystać 0 lub więcej klientów
Klienci – Adresy	KlientPosiadaAdres	jeden do jeden	każdy klient posiada dokładnie jeden adres
Klienci – Bilety	KorzystaKlient	jeden do wielu	co najmniej jeden klient może korzystać z wielu biletów; jeden bilet może należeć do co najwyżej dokładnie jednego klienta
Atrakcje – Bilety	KorzystaAtrakcja	wiele do wielu	1 lub więcej atrakcji może być przypisane do 1 lub więcej biletów; jeden bilet może być przypisany do 1 lub więcej atrakcji

Tabela 2: Opis relacji między encjami w kontekście parku rozrywki (cz2).

Określenie atrybutów i ich dziedzin:

Tabela 3: Atrybuty parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idParku	Integer	Tak	Unikatowy numer parku rozrywki
nazwa	VarChar(30)	Tak	Nazwa parku rozrywki
nrTelefonu	VarChar(15)	Tak	Główny numer telefonu do recepcji parku rozrywki
email	VarChar(30)	Tak	Główny email do recepcji parku rozrywki

Tabela 4: Atrybuty właściciela parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idWlasciciela	Integer	Tak	Unikatowy numer właściciela parku rozrywki
imie	VarChar(20)	Tak	Imię właściciela parku rozrywki
nazwisko	VarChar(20)	Tak	Nazwisko właściciela parku rozrywki

Tabela 5: Atrybuty pracownika parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idPracownika	Integer	Tak	Unikatowy numer pracownika parku rozrywki
imie	VarChar(20)	Tak	Imię pracownika parku
drugieImie	VarChar(20)	Nie	Nieobowiązkowe drugie imię pracownika parku
nazwisko	VarChar(30)	Tak	Nazwisko pracownika parku
dataUrodzin	DateTime2	Tak	Data urodzin pracownika parku
PESEL	Char(11)	Nie	Numer PESEL pracownika parku
plec	Char(1) ('K', 'M')	Tak	Płeć pracownika parku
dataZatrudnienia	DateTime2	Tak	Data zatrudnienia pracownika w parku
nrTelefonu	VarChar(15)	Tak	Numer telefonu pracownika parku rozrywki

Tabela 6: Atrybuty wynagrodzeń dla pracowników parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idWynagrodzenia	Integer	Tak	Unikatowy numer wynagrodzenia dla pracownika parku
data	DateTime2	Tak	Data wypłacenia wynagrodzenia
kwotaPod	Decimal(10,2)	Tak	Kwota podstawowa wypłaconego wynagrodzenia
kwotaDod	Decimal(10,2)	Nie	Kwota dodatkowa wypłaconego wynagrodzenia

Tabela 7: Atrybuty punktu sprzedaży w parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idPunktuSprzedaży	Integer	Tak	Unikatowy numer punktu sprzedaży w parku rozrywki
liczbaKas	Integer	Tak	Liczba kas punktu sprzedaży

Tabela 8: Atrybuty stanowiska pracownika parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idStanowiska	Integer	Tak	Unikatowy numer stanowiska pracownika parku rozrywki
nazwa	VarChar(20)	Tak	Nazwa stanowiska pracownika parku rozrywki
opis	VarChar(400)	Tak	Opis stanowiska pracownika parku rozrywki

Tabela 9: Atrybuty adresu.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idAdresu	Integer	Tak	Unikatowy numer adresu
kraj	VarChar(40)	Tak	Nazwa kraju (w języku polskim)
miasto	VarChar(30)	Tak	Nazwa miasta (w języku polskim)
ulica	VarChar(30)	Tak	Nazwa ulicy (w języku polskim)
nrBudynku	VarChar(5)	Tak	Numer budynku na ulicy
nrLokalu	VarChar(5)	Nie	Numer lokalu w budynku



Tabela 10: Atrybuty poczty adresu.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idPoczty	Integer	Tak	Unikatowy numer poczty
kodPocztowy	Char(6)	Tak	Kod pocztowy danej poczty
poczta	VarChar(20)	Tak	Nazwa poczty

Tabela 11: Atrybuty usługi w parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idUsługi	Integer	Tak	Unikatowy numer usługi w parku rozrywki
nazwa	VarChar(30)	Tak	Nazwa usługi w parku rozrywki
opis	VarChar(400)	Tak	Opis usługi w parku rozrywki

Tabela 12: Atrybuty atrakcji w parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idAtrakcji	Integer	Tak	Unikatowy numer atrakcji w parku rozrywki
nazwa	VarChar(20)	Tak	Nazwa atrakcji w parku rozrywki
minWzrost	Integer	Tak	Minimalny wzrost potrzebny do korzystania z atrakcji przez klienta
dlugoscTrasy	Integer	Nie	Długość trasy danej atrakcji
dataPrzeglądu	DateTime2	Tak	Data ostatniego przeglądu atrakcji

Tabela 13: Atrybuty biletu w parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idBiletu	Integer	Tak	Unikatowy numer biletu w parku rozrywki
cena	Decimal(10,2)	Tak	Cena biletu
dataWaznosci	DateTime2	Tak	Data ważności biletu
dataZakupu	DateTime2	Tak	Data zakupu biletu
czyZakupionyStacjonarnie	Bit	Tak	Określa, czy bilet został zakupiony stacjonarnie
czyWykorzystany	Bit	Nie	Określa, czy bilet był już wykorzystany

Tabela 14: Atrybuty klienta parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idKlienta	Integer	Tak	Unikatowy numer klienta parku rozrywki
nrTelefonu	VarChar(15)	Tak	Numer telefonu klienta wymagany przy zakupie biletów
email	VarChar(30)	Tak	Adres e-mail klienta wymagany przy zakupie biletów

Tabela 15: Atrybuty klienta-szkoły parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idKlienta	Integer	Tak	Unikatowy numer klienta parku rozrywki
nazwa	VarChar(50)	Tak	Nazwa szkoły
NIP	VarChar(10)	Tak	Numer NIP szkoły
REGON	VarChar(9)	Tak	Numer REGON szkoły

Tabela 16: Atrybuty klienta-osoby prywatnej parku rozrywki.

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
idKlienta	Integer	Tak	Unikatowy numer klienta parku rozrywki
imie	VarChar(20)	Tak	Imię klienta parku rozrywki
drugieImie	VarChar(20)	Nie	Nieobowiązkowe drugie imię klienta
nazwisko	VarChar(20)	Tak	Nazwisko klienta parku rozrywki
wzrost	Integer	Nie	Wzrost klienta
dataUrodzenia	DateTime2	Tak	Data urodzenia klienta
plec	Char(1) ('K', 'M')	Nie	Płeć klienta

### 2.3. Klucze kandydujące i główne

Nazwa encji	Klucz główny	Klucz kandydujący
ParkiRozrywki	idParku	nrTelefonu, email
Wlasciciele	idWlasciciela	-
Pracownicy	idPracownika	nrTelefonu
Wynagrodzenia	idWynagrodzenia	-
PunktySprzedazy	idPunktuSprzedazy	-
Stanowiska	idStanowiska	nazwa
Adresy	idAdresu	-
Poczty	idPoczty	kodPocztowy, poczta
Uslugi	idUslugi	nazwa
Bilety	idBiletu	-
Klienci	idKlienta	nrTelefonu, email

Tabela 17: Struktura encji, klucze główne i kandydujące

### 2.4. Postać normalna

#### Postać Normalna - Kryterium pierwszej postaci normalnej:

Aby doprowadzić bazę danych do pierwszej postaci normalnej, należy przestrzegać następujących reguł:

- Każda wartość atrybutu w każdej krotce relacji jest wartością elementarną (atomową), czyli taką, której się nie da rozłożyć.
- Brak powtarzających się grup.

#### Postać Normalna - Kryterium drugiej postaci normalnej:

Kryterium drugiej postaci normalnej zakłada, że pierwsza postać normalna została spełniona oraz wszystkie atrybuty w tabeli, które nie są kluczami, są w pełni funkcyjne i zależne od wszystkich innych kluczy w relacji. Klucz powinien być prosty. W projekcie każdy klucz jest kluczem prostym - identyfikatorem ID.

#### Postać Normalna - Kryterium trzeciej postaci normalnej:

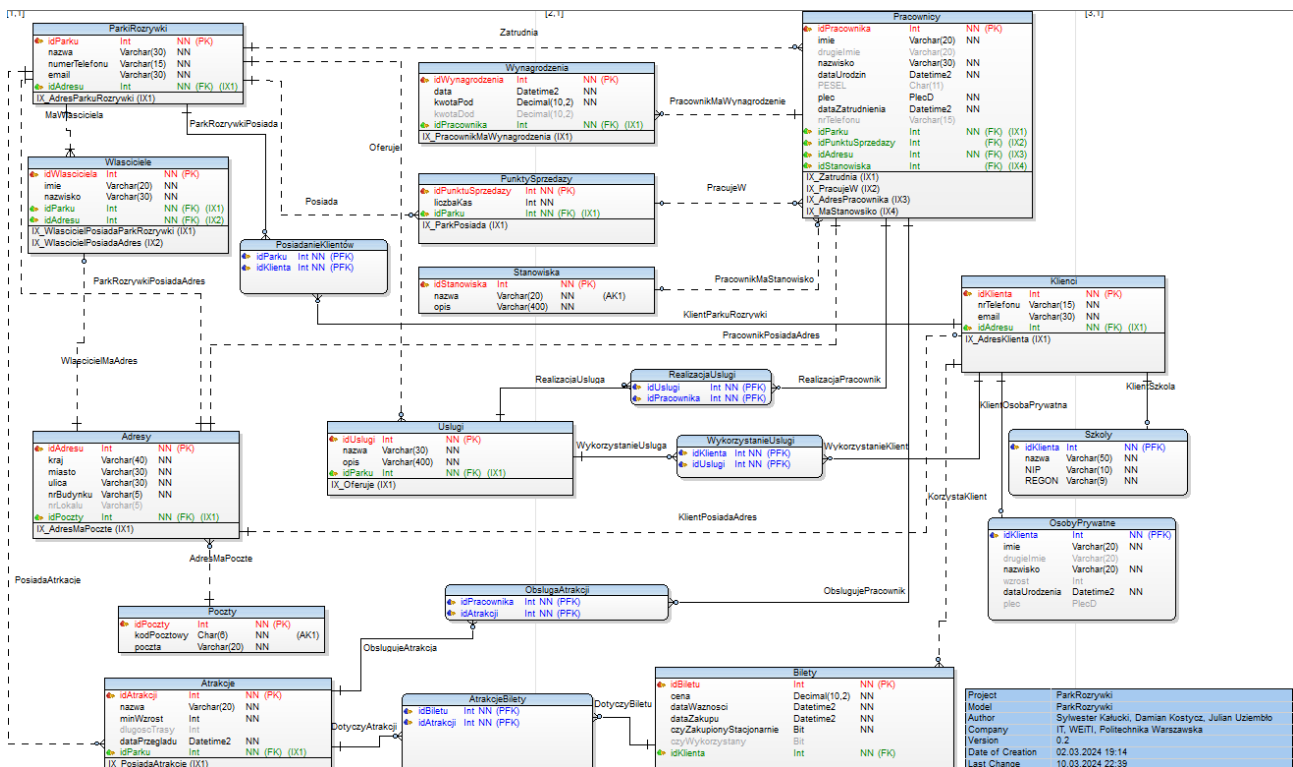
Kryterium trzeciej postaci normalnej zakłada, że druga postać normalna została spełniona oraz żaden atrybut kluczowy nie jest zależny funkcyjnie od innych atrybutów niekluczowych.

## Więzy integralności:

Więzy integralności jest to zgodność pomiędzy zawartością pól rekordów oraz typami, jakie mogą przyjmować. Dla przykładu: jeśli dla atrybutu “imie” jego typ został określony jako VarChar(20) to pole to powinno przechowywać taką właśnie wartość, a nie, na przykład Integer.

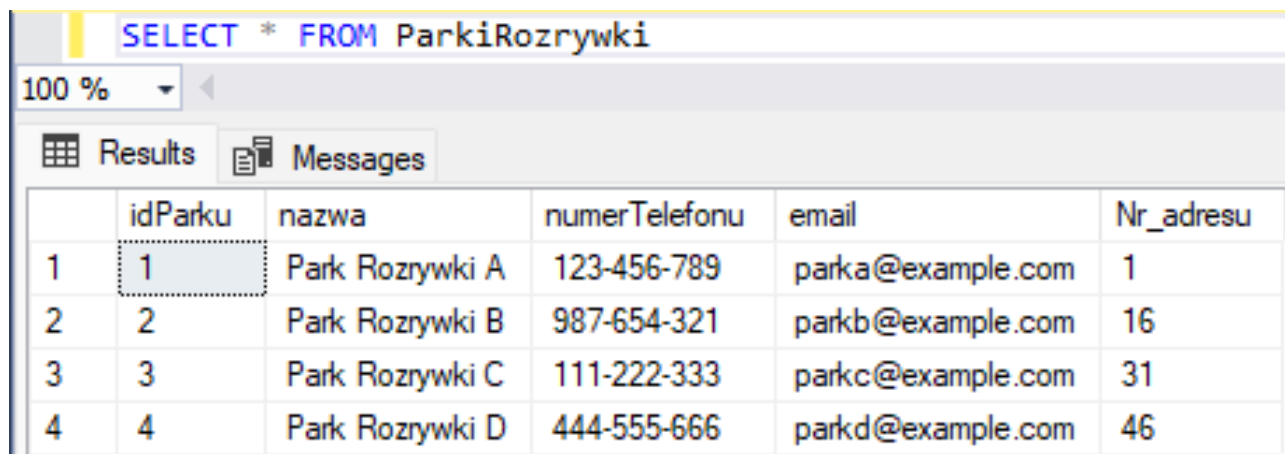
Aby spełnić warunki więzów integralności każde pole jest polem atomowym (co zostało dokonane podczas 1 postaci normalnej), każda encja ma jeden unikatowy klucz główny, a problem, że dany atrybut przyjmuje wartość NULL zdarza się niezwykle rzadko, gdyż większość pól jest obowiązkowa.

## 2.5. Schemat modelu fizycznego



Rysunek 1: Schemat modelu fizycznego bazy Parki rozrywki.

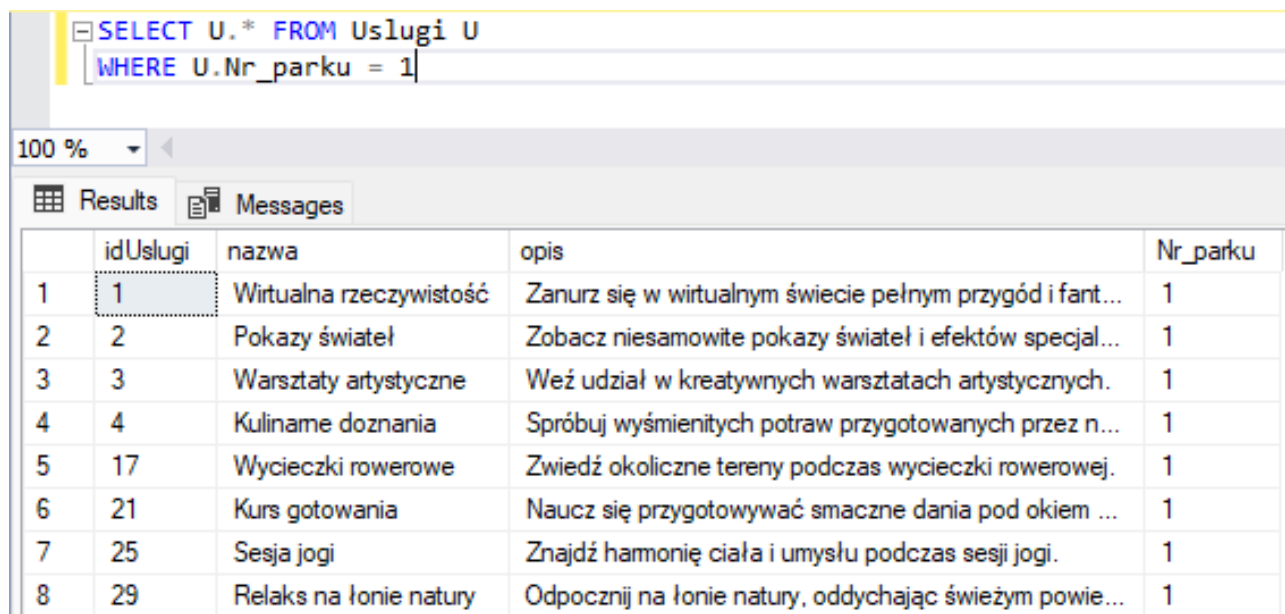
## 2.6. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych



```
SELECT * FROM ParkiRozrywki
```

	idParku	nazwa	numerTelefonu	email	Nr_adresu
1	1	Park Rozrywki A	123-456-789	parka@example.com	1
2	2	Park Rozrywki B	987-654-321	parkb@example.com	16
3	3	Park Rozrywki C	111-222-333	parkc@example.com	31
4	4	Park Rozrywki D	444-555-666	parkd@example.com	46

Rysunek 2: Wybierz wszystkie informacje o parkach rozrywki



```
SELECT U.* FROM Usługi U  
WHERE U.Nr_parku = 1
```

	idUsługi	nazwa	opis	Nr_parku
1	1	Wirtualna rzeczywistość	Zanurz się w wirtualnym świecie pełnym przygód i fant...	1
2	2	Pokazy świateł	Zobacz niesamowite pokazy świateł i efektów specjal...	1
3	3	Warsztaty artystyczne	Weź udział w kreatywnych warsztatach artystycznych.	1
4	4	Kulinarne doznania	Spróbuj wyśmienitych potraw przygotowanych przez n...	1
5	17	Wycieczki rowerowe	Zwiedź okoliczne tereny podczas wycieczki rowerowej.	1
6	21	Kurs gotowania	Naucz się przygotowywać smaczne dania pod okiem ...	1
7	25	Sesja jogi	Znajdź harmonię ciała i umysłu podczas sesji jogi.	1
8	29	Relaks na łonie natury	Odpocznij na łonie natury, oddychając świeżym powie...	1

Rysunek 3: Znajdź dodatkowe usługi świadczone w parku nr 1

SELECT * FROM Bilety WHERE czyZakupionyStacjonarnie = 1 AND czyWykorzystany = 0;							
100 %							
Results Messages							
	idBiletu	cena	dataWaznosci	dataZakupu	czyZakupionyStacjonarnie	czyWykorzystany	Nr_klienta
1	1	150.75	2024-04-30 00:00:00.0000000	2023-07-01 00:00:00.0000000	1	0	1
2	2	120.25	2024-05-15 00:00:00.0000000	2023-08-10 00:00:00.0000000	1	0	2
3	4	210.30	2024-04-25 00:00:00.0000000	2023-10-20 00:00:00.0000000	1	0	4
4	6	180.90	2024-04-15 00:00:00.0000000	2023-12-05 00:00:00.0000000	1	0	6
5	7	147.25	2024-04-30 00:00:00.0000000	2024-01-15 00:00:00.0000000	1	0	7
6	9	160.40	2024-04-10 00:00:00.0000000	2024-03-01 00:00:00.0000000	1	0	9
7	11	212.70	2024-05-20 00:00:00.0000000	2024-05-01 00:00:00.0000000	1	0	11
8	12	197.90	2024-04-15 00:00:00.0000000	2024-06-10 00:00:00.0000000	1	0	12
9	14	178.80	2024-04-20 00:00:00.0000000	2024-08-20 00:00:00.0000000	1	0	14
10	16	220.20	2024-05-15 00:00:00.0000000	2024-10-10 00:00:00.0000000	1	0	16
11	17	169.90	2024-04-10 00:00:00.0000000	2024-11-15 00:00:00.0000000	1	0	17
12	19	143.60	2024-04-25 00:00:00.0000000	2025-01-01 00:00:00.0000000	1	0	19
13	21	165.80	2024-05-15 00:00:00.0000000	2025-03-15 00:00:00.0000000	1	0	21
14	22	211.25	2024-04-20 00:00:00.0000000	2025-04-20 00:00:00.0000000	1	0	22
15	24	189.90	2024-04-15 00:00:00.0000000	2025-06-10 00:00:00.0000000	1	0	24
16	27	194.90	2024-04-10 00:00:00.0000000	2025-09-01 00:00:00.0000000	1	0	27
17	28	162.10	2024-05-20 00:00:00.0000000	2025-10-10 00:00:00.0000000	1	0	28
18	30	146.40	2024-05-10 00:00:00.0000000	2025-12-20 00:00:00.0000000	1	0	30
19	32	199.90	2024-05-05 00:00:00.0000000	2026-02-10 00:00:00.0000000	1	0	32
20	35	115.30	2024-04-25 00:00:00.0000000	2026-05-01 00:00:00.0000000	1	0	35
21	36	188.80	2024-05-10 00:00:00.0000000	2026-06-10 00:00:00.0000000	1	0	36
22	38	224.90	2024-04-20 00:00:00.0000000	2026-08-20 00:00:00.0000000	1	0	38
23	39	198.75	2024-05-05 00:00:00.0000000	2026-09-01 00:00:00.0000000	1	0	39
24	41	4050.00	2024-11-30 00:00:00.0000000	2023-07-01 00:00:00.0000000	1	0	41
25	42	4200.00	2024-12-15 00:00:00.0000000	2023-08-10 00:00:00.0000000	1	0	42
26	44	4500.00	2024-11-25 00:00:00.0000000	2023-10-20 00:00:00.0000000	1	0	44
27	46	4800.00	2024-11-15 00:00:00.0000000	2023-12-05 00:00:00.0000000	1	0	46
28	47	4950.00	2024-11-30 00:00:00.0000000	2024-01-15 00:00:00.0000000	1	0	47
29	49	4700.00	2024-11-10 00:00:00.0000000	2024-03-01 00:00:00.0000000	1	0	49

Rysunek 4: Pokaż wszystkie bilety, które zostały zakupione stacjonarnie i nie zostały jeszcze wykorzystane

```

SELECT
    A.miasto AS Miasto,
    COUNT(K.idKlienta) AS LiczbaKlientow
FROM
    Adresy A
JOIN
    Klienci K ON A.idAdresu = K.Nr_adresu
GROUP BY
    A.miasto
ORDER BY
    LiczbaKlientow DESC;

```

	Miasto	LiczbaKlientow
1	Gdańsk	26
2	Kraków	14
3	Wrocław	10

Rysunek 5: Ilość klientów z poszczególnych miast uszeregowana od największej liczby

```

SELECT
    P.imie,
    P.nazwisko,
    W.data AS DataWynagrodzenia,
    W.kwotaPod AS KwotaPodstawowa,
    W.kwotaDod AS DodatkoweSkładki,
    (W.kwotaPod + ISNULL(W.kwotaDod, 0)) AS CalkowitaKwota
FROM
    Pracownicy P
JOIN
    Wynagrodzenia W ON P.idPracownika = W.Nr_pracownika
ORDER BY
    P.idPracownika, W.data;

```

	imie	nazwisko	DataWynagrodzenia	KwotaPodstawowa	DodatkoweSkładki	CalkowitaKwota
1	Adam	Nowak	2024-03-01 00:00:00.0000000	3000.00	200.00	3200.00
2	Ewa	Kowalska	2024-03-01 00:00:00.0000000	2800.00	150.00	2950.00
3	Jan	Wojciechowski	2024-03-01 00:00:00.0000000	3200.00	180.00	3380.00
4	Karolina	Duda	2024-03-01 00:00:00.0000000	2900.00	160.00	3060.00
5	Piotr	Lis	2024-03-01 00:00:00.0000000	3100.00	190.00	3290.00
6	Alicja	Kaczmarek	2024-03-01 00:00:00.0000000	3000.00	NULL	3000.00
7	Marek	Wolski	2024-03-01 00:00:00.0000000	2900.00	170.00	3070.00
8	Anna	Nowak	2024-03-01 00:00:00.0000000	3100.00	200.00	3300.00
9	Krzysztof	Kowal	2024-03-01 00:00:00.0000000	2800.00	140.00	2940.00
10	Wiktoria	Piotrowska	2024-03-01 00:00:00.0000000	3300.00	210.00	3510.00
11	Bartosz	Kowalczyk	2024-03-01 00:00:00.0000000	3000.00	180.00	3180.00
12	Dominika	Lis	2024-03-01 00:00:00.0000000	2900.00	160.00	3060.00
13	Michał	Pawlak	2024-03-01 00:00:00.0000000	3100.00	190.00	3290.00
14	Karolina	Zielińska	2024-03-01 00:00:00.0000000	3000.00	NULL	3000.00
15	Rafał	Nowicki	2024-03-01 00:00:00.0000000	2800.00	150.00	2950.00
16	Sylvia	Szczepańska	2024-03-01 00:00:00.0000000	3200.00	180.00	3380.00
17	Paweł	Olszewski	2024-03-01 00:00:00.0000000	2900.00	160.00	3060.00
18	Natalia	Jankowska	2024-03-01 00:00:00.0000000	3100.00	190.00	3290.00
19	Sebastian	Witkowski	2024-03-01 00:00:00.0000000	3000.00	NULL	3000.00
20	Monika	Głowacka	2024-03-01 00:00:00.0000000	2900.00	170.00	3070.00

Rysunek 6: Koszty pracownicze

## 2.7. Skrypt w SQL

```
/*
Created: 02.03.2024
Modified: 10.03.2024
Project: Park_rozrywki
Model: Park_Rozrywki
Company: IT, WEiTl, Politechnika Warszawska
Author: Sylwester Kałucki, Damian Kostycz, Julian Uziembło
Version: 0.2
Database: MS SQL Server 2017
*/

-- Create rules section -----

CREATE RULE [PlecR] AS @Plec IN ('K','M')
go

-- Create tables section -----

-- Table ParkiRozrywki

CREATE TABLE [ParkiRozrywki]
(
    [idParku] Int NOT NULL,
    [nazwa] Varchar(30) NOT NULL,
    [numerTelefonu] Varchar(15) NOT NULL,
    [email] Varchar(30) NOT NULL,
    [idAdresu] Int NOT NULL
)
go

-- Create indexes for table ParkiRozrywki

CREATE INDEX [IX_AdresParkuRozrywki] ON [ParkiRozrywki] ([idAdresu])
go

-- Add keys for table ParkiRozrywki

ALTER TABLE [ParkiRozrywki] ADD CONSTRAINT [ParkRozrywkiPK] PRIMARY KEY ([idParku])
go

-- Table Pracownicy

CREATE TABLE [Pracownicy]
(
    [idPracownika] Int NOT NULL,
    [imie] Varchar(20) NOT NULL,
    [drugieImie] Varchar(20) NULL,
    [nazwisko] Varchar(30) NOT NULL,
    [dataUrodzin] Datetime2 NOT NULL,
    [PESEL] Char(11) NULL,
    [plec] Char(256) NOT NULL
        CHECK (@Plec IN ('K','M')),
    [dataZatrudnienia] Datetime2 NOT NULL,
    [nrTelefonu] Varchar(15) NULL,
    [idParku] Int NOT NULL,
    [idPunktuSprzedazy] Int NULL,
    [idAdresu] Int NOT NULL,
    [idStanowiska] Int NULL
)
```



```

)
go

-- Create indexes for table Pracownicy

CREATE INDEX [IX_Zatrudnia] ON [Pracownicy] ([idParku])
go

CREATE INDEX [IX_PracujeW] ON [Pracownicy] ([idPunktuSprzedazy])
go

CREATE INDEX [IX_AdresPracownika] ON [Pracownicy] ([idAdresu])
go

CREATE INDEX [IX_MaStanowsiko] ON [Pracownicy] ([idStanowiska])
go

-- Add keys for table Pracownicy

ALTER TABLE [Pracownicy] ADD CONSTRAINT [PracownikPK] PRIMARY KEY ([idPracownika])
go

-- Table OsobyPrywatne

CREATE TABLE [OsobyPrywatne]
(
    [idKlienta] Int NOT NULL,
    [imie] Varchar(20) NOT NULL,
    [drugieImie] Varchar(20) NULL,
    [nazwisko] Varchar(20) NOT NULL,
    [wzrost] Int NULL,
    [dataUrodzenia] Datetime2 NOT NULL,
    [plec] Char(256) NULL
        CHECK (@Plec IN ('K','M'))
)
go

-- Add keys for table OsobyPrywatne

ALTER TABLE [OsobyPrywatne] ADD CONSTRAINT [Osoby_prywatne1] PRIMARY KEY ([idKlienta])
go

-- Table Atrakcje

CREATE TABLE [Atrakcje]
(
    [idAtrakcji] Int NOT NULL,
    [nazwa] Varchar(20) NOT NULL,
    [minWzrost] Int NOT NULL,
    [dlugoscTrasy] Int NULL,
    [dataPrzeglądu] Datetime2 NOT NULL,
    [idParku] Int NOT NULL
)
go

-- Create indexes for table Atrakcje

CREATE INDEX [IX_PosiadaAtrakcje] ON [Atrakcje] ([idParku])
go

-- Add keys for table Atrakcje

```

```

ALTER TABLE [Atrakcje] ADD CONSTRAINT [AtrakcjaPK] PRIMARY KEY ([idAtrakcji])
go

-- Table Klienci

CREATE TABLE [Klienci]
(
    [idKlienta] Int NOT NULL,
    [nrTelefonu] Varchar(15) NOT NULL,
    [email] Varchar(30) NOT NULL,
    [idAdresu] Int NOT NULL
)
go

-- Create indexes for table Klienci

CREATE INDEX [IX_AdresKlienta] ON [Klienci] ([idAdresu])
go

-- Add keys for table Klienci

ALTER TABLE [Klienci] ADD CONSTRAINT [KlientPK] PRIMARY KEY ([idKlienta])
go

-- Table Szkoly

CREATE TABLE [Szkoly]
(
    [idKlienta] Int NOT NULL,
    [nazwa] Varchar(50) NOT NULL,
    [NIP] Varchar(10) NOT NULL,
    [REGON] Varchar(9) NOT NULL
)
go

-- Add keys for table Szkoly

ALTER TABLE [Szkoly] ADD CONSTRAINT [Unique_Identifier2] PRIMARY KEY ([idKlienta])
go

-- Table PunktySprzedazy

CREATE TABLE [PunktySprzedazy]
(
    [idPunktuSprzedazy] Int NOT NULL,
    [liczbaKas] Int NOT NULL,
    [idParku] Int NOT NULL
)
go

-- Create indexes for table PunktySprzedazy

CREATE INDEX [IX_ParkPosiada] ON [PunktySprzedazy] ([idParku])
go

-- Add keys for table PunktySprzedazy

ALTER TABLE [PunktySprzedazy] ADD CONSTRAINT [PunktSprzedarzyPK] PRIMARY KEY ([idPunktuSprzedazy])
go

```

```

-- Table Bilety

CREATE TABLE [Bilety]
(
    [idBiletu] Int NOT NULL,
    [cena] Decimal(10,2) NOT NULL,
    [dataWaznosci] Datetime2 NOT NULL,
    [dataZakupu] Datetime2 NOT NULL,
    [czyZakupionyStacjonarnie] Bit NOT NULL,
    [czyWykorzystany] Bit DEFAULT 0 NULL,
    [idKlienta] Int NOT NULL
)
go

-- Add keys for table Bilety

ALTER TABLE [Bilety] ADD CONSTRAINT [BiletyPK] PRIMARY KEY ([idBiletu])
go

-- Table ObslugaAtrakcji

CREATE TABLE [ObslugaAtrakcji]
(
    [idPracownika] Int NOT NULL,
    [idAtrakcji] Int NOT NULL
)
go

-- Table Adresy

CREATE TABLE [Adresy]
(
    [idAdresu] Int NOT NULL,
    [kraj] Varchar(40) NOT NULL,
    [miasto] Varchar(30) NOT NULL,
    [ulica] Varchar(30) NOT NULL,
    [nrBudynku] Varchar(5) NOT NULL,
    [nrLokalu] Varchar(5) NULL,
    [idPoczty] Int NOT NULL
)
go

-- Create indexes for table Adresy

CREATE INDEX [IX_AdresMaPoczte] ON [Adresy] ([idPoczty])
go

-- Add keys for table Adresy

ALTER TABLE [Adresy] ADD CONSTRAINT [AdresyPK] PRIMARY KEY ([idAdresu])
go

-- Table Wlasciciele

CREATE TABLE [Wlasciciele]
(
    [idWlasciciela] Int NOT NULL,
    [imie] Varchar(20) NOT NULL,
    [nazwisko] Varchar(30) NOT NULL,
    [idParku] Int NOT NULL,

```

```

[idAdresu] Int NOT NULL
)
go

-- Create indexes for table Wlasciciele

CREATE INDEX [IX_WlascicielPosiadaParkRozrywki] ON [Wlasciciele] ([idParku])
go

CREATE INDEX [IX_WlascicielPosiadaAdres] ON [Wlasciciele] ([idAdresu])
go

-- Add keys for table Wlasciciele

ALTER TABLE [Wlasciciele] ADD CONSTRAINT [WlascicielePK] PRIMARY KEY ([idWlasciciela])
go

-- Table Stanowiska

CREATE TABLE [Stanowiska]
(
    [idStanowiska] Int NOT NULL,
    [nazwa] Varchar(20) NOT NULL,
    [opis] Varchar(400) NOT NULL
)
go

-- Add keys for table Stanowiska

ALTER TABLE [Stanowiska] ADD CONSTRAINT [StanowiskaPK] PRIMARY KEY NONCLUSTERED ([idStanowiska])
go

ALTER TABLE [Stanowiska] ADD CONSTRAINT [Nazwa] UNIQUE CLUSTERED ([nazwa])
go

-- Table Wynagrodzenia

CREATE TABLE [Wynagrodzenia]
(
    [idWynagrodzenia] Int NOT NULL,
    [data] Datetime2 NOT NULL,
    [kwotaPod] Decimal(10,2) NOT NULL,
    [kwotaDod] Decimal(10,2) NULL,
    [idPracownika] Int NOT NULL
)
go

-- Create indexes for table Wynagrodzenia

CREATE INDEX [IX_PracownikMaWynagrodzenia] ON [Wynagrodzenia] ([idPracownika])
go

-- Add keys for table Wynagrodzenia

ALTER TABLE [Wynagrodzenia] ADD CONSTRAINT [PK_Wynagrodzenia] PRIMARY KEY ([idWynagrodzenia])
go

-- Table Poczty

CREATE TABLE [Poczty]
(

```

```

[idPoczty] Int NOT NULL,
[kodPocztowy] Char(6) NOT NULL,
[poczta] Varchar(20) NOT NULL
)
go

-- Add keys for table Poczty

ALTER TABLE [Poczty] ADD CONSTRAINT [PocztyPK] PRIMARY KEY NONCLUSTERED ([idPoczty])
go

ALTER TABLE [Poczty] ADD CONSTRAINT [kodPocztowy] UNIQUE CLUSTERED ([kodPocztowy])
go

-- Table Uslugi

CREATE TABLE [Uslugi]
(
[idUslugi] Int NOT NULL,
[nazwa] Varchar(30) NOT NULL,
[opis] Varchar(400) NOT NULL,
[idParku] Int NOT NULL
)
go

-- Create indexes for table Uslugi

CREATE INDEX [IX_Oferuje] ON [Uslugi] ([idParku])
go

-- Add keys for table Uslugi

ALTER TABLE [Uslugi] ADD CONSTRAINT [UslugiPK] PRIMARY KEY NONCLUSTERED ([idUslugi])
go

-- Table WykorzystanieUslugi

CREATE TABLE [WykorzystanieUslugi]
(
[idKlienta] Int NOT NULL,
[idUslugi] Int NOT NULL
)
go

-- Add keys for table WykorzystanieUslugi

ALTER TABLE [WykorzystanieUslugi] ADD CONSTRAINT
[PK_WykorzystanieUslugi] PRIMARY KEY ([idKlienta],[idUslugi])
go

-- Table RealizacjaUslugi

CREATE TABLE [RealizacjaUslugi]
(
[idUslugi] Int NOT NULL,
[idPracownika] Int NOT NULL
)
go

-- Add keys for table RealizacjaUslugi

```

```

ALTER TABLE [RealizacjaUslugi] ADD CONSTRAINT
[PK_RealizacjaUslugi] PRIMARY KEY ([idUslugi],[idPracownika])
go

-- Table PosiadanieKlientów

CREATE TABLE [PosiadanieKlientów]
(
    [idParku] Int NOT NULL,
    [idKlienta] Int NOT NULL
)
go

-- Add keys for table PosiadanieKlientów

ALTER TABLE [PosiadanieKlientów] ADD CONSTRAINT
[PK_PosiadanieKlientów] PRIMARY KEY ([idParku],[idKlienta])
go

-- Table AtrakcjeBilety

CREATE TABLE [AtrakcjeBilety]
(
    [idBiletu] Int NOT NULL,
    [idAtrakcji] Int NOT NULL
)
go

-- Add keys for table AtrakcjeBilety

ALTER TABLE [AtrakcjeBilety] ADD CONSTRAINT
[PK_AtrakcjeBilety] PRIMARY KEY ([idBiletu],[idAtrakcji])
go

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

ALTER TABLE [Pracownicy] ADD CONSTRAINT [Zatrudnia]
FOREIGN KEY ([idParku]) REFERENCES [ParkiRozrywki] ([idParku])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [Atrakcje] ADD CONSTRAINT [PosiadaAtrkacje]
FOREIGN KEY ([idParku]) REFERENCES [ParkiRozrywki] ([idParku])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [PunktySprzedazy] ADD CONSTRAINT [Posiada]
FOREIGN KEY ([idParku]) REFERENCES [ParkiRozrywki] ([idParku])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [Pracownicy] ADD CONSTRAINT [PracujeW]
FOREIGN KEY ([idPunktuSprzedazy]) REFERENCES [PunktySprzedazy] ([idPunktuSprzedazy])
ON UPDATE NO ACTION ON DELETE NO ACTION

```

go

```
ALTER TABLE [ParkiRozrywki] ADD CONSTRAINT [ParkRozrywkiPosiadaAdres]
FOREIGN KEY ([idAdresu]) REFERENCES [Adresy] ([idAdresu])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Pracownicy] ADD CONSTRAINT [PracownikPosiadaAdres]
FOREIGN KEY ([idAdresu]) REFERENCES [Adresy] ([idAdresu])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Klienci] ADD CONSTRAINT [KlientPosiadaAdres]
FOREIGN KEY ([idAdresu]) REFERENCES [Adresy] ([idAdresu])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Wlasciciele] ADD CONSTRAINT [MaWlasciciela]
FOREIGN KEY ([idParku]) REFERENCES [ParkiRozrywki] ([idParku])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Wlasciciele] ADD CONSTRAINT [WlascicielMaAdres]
FOREIGN KEY ([idAdresu]) REFERENCES [Adresy] ([idAdresu])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Pracownicy] ADD CONSTRAINT [PracownikMaStanowisko]
FOREIGN KEY ([idStanowiska]) REFERENCES [Stanowiska] ([idStanowiska])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Wynagrodzenia] ADD CONSTRAINT [PracownikMaWynagrodzenie]
FOREIGN KEY ([idPracownika]) REFERENCES [Pracownicy] ([idPracownika])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Adresy] ADD CONSTRAINT [AdresMaPoczte]
FOREIGN KEY ([idPoczty]) REFERENCES [Poczty] ([idPoczty])
ON UPDATE NO ACTION ON DELETE NO ACTION
```

go

```
ALTER TABLE [Uslugi] ADD CONSTRAINT [Ofaruje]
```

```

FOREIGN KEY ([idParku]) REFERENCES [ParkiRozrywki] ([idParku])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [WykorzystanieUslugi] ADD CONSTRAINT [WykorzystanieKlient]
FOREIGN KEY ([idKlienta]) REFERENCES [Klienci] ([idKlienta])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [WykorzystanieUslugi] ADD CONSTRAINT [WykorzystanieUslugi]
FOREIGN KEY ([idUslugi]) REFERENCES [Uslugi] ([idUslugi])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [RealizacjaUslugi] ADD CONSTRAINT [RealizacjaUslugi]
FOREIGN KEY ([idUslugi]) REFERENCES [Uslugi] ([idUslugi])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [RealizacjaUslugi] ADD CONSTRAINT [RealizacjaPracownik]
FOREIGN KEY ([idPracownika]) REFERENCES [Pracownicy] ([idPracownika])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [PosiadanieKlientów] ADD CONSTRAINT [ParkRozrywkiPosiada]
FOREIGN KEY ([idParku]) REFERENCES [ParkiRozrywki] ([idParku])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [PosiadanieKlientów] ADD CONSTRAINT [KlientParkuRozrywki]
FOREIGN KEY ([idKlienta]) REFERENCES [Klienci] ([idKlienta])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [AtrakcjeBilety] ADD CONSTRAINT [DotyczyAtrakcji]
FOREIGN KEY ([idAtrakcji]) REFERENCES [Atrakcje] ([idAtrakcji])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE [AtrakcjeBilety] ADD CONSTRAINT [DotyczyBiletu]
FOREIGN KEY ([idBiletu]) REFERENCES [Bilety] ([idBiletu])
ON UPDATE NO ACTION ON DELETE NO ACTION
go

```



### 3. Baza relacyjna - Akcje Marketingowe

#### 3.1. Definicja systemu

W naszym systemie wyróżniamy następujące funkcjonalności:

- podgląd informacji o akcji marketingowej,
- dodawanie/modyfikowanie/usuwanie informacji o akcjach marketingowych
- podgląd informacji o szczegółach akcji marketingowych
- dodawanie/modyfikowanie/usuwanie informacji o szczegółach akcji marketingowych
- podgląd informacji o kanałach akcji,
- dodawanie/modyfikowanie/usuwanie informacji o kanałach akcji
- podgląd informacji o uczestnikach akcji,
- dodawanie/modyfikowanie/usuwanie informacji o uczestnikach akcji
- podgląd informacji o nagrodach,
- dodawanie/modyfikowanie/usuwanie informacji o nagrodach

#### 3.2. Encje, związki oraz atrybuty

Definicja zbiorów encji określonych w projekcie:

- AkcjeMarketingowe, główna encja opisująca akcje marketingowe, których dane znajdują się w bazie,
- AkcjeSzczegoly, encja zawierająca dane ogólne o szczegółach akcji marketingowych
- KanalyAkcji, encja zawierająca dane ogólne o kanałach akcji
- UczestnicyAkcji, encja zawierająca dane ogólne o uczestnikach akcji
- Nagrody, encja zawierająca dane ogólne o nagrodach dla uczestników

Tabela 18: Opis relacji między encjami w kontekście parku rozrywki.

Nazwa relacji	Krotność	Typ uczestnictwa	Opis
AkcjeMarketingowe – AkcjeSzczegoly	jeden do wielu	akcja marketingowa jest obowiązkowy dla szczegółów akcji; szczegół akcji jest opcjonalny dla akcji marketingowej	Akcja marketingowa może mieć wiele szczegółów, ale może być na etapie otwierania się, czyli na etapie gdzie nie ma żadnego szczegółu. Szczegół aby należeć do akcji musi brać udział w jednej akcji marketingowej.
AkcjeMarketingowe – KanalyAkcji	jeden do wielu	akcja marketingowa jest obowiązkowy dla kanałów akcji; kanał akcji jest opcjonalny dla akcji marketingowej	Akcja marketingowa może mieć wiele kanałów, ale może być na etapie otwierania się, czyli na etapie gdzie może nie mieć żadnego kanału. Kanał, aby być częścią akcji marketingowej, musi znajdować się w jednej akcji marketingowej.
AkcjeMarketingowe – UczestnicyAkcji	wielu do wielu	akcja marketingowa jest obowiązkowa dla uczestników akcji; uczestnik może korzystać z wielu akcji, z akcji może skorzystać wiele uczestników	akcja marketingowa może mieć wielu uczestników ale może też nie mieć żadnego
UczestnicyAkcji – Nagrody	jeden do wielu	uczestnik akcji jest obowiązkowy dla nagrody; nagroda jest opcjonalna dla uczestnika	uczestnik może mieć wiele nagród, ale może również nie mieć żadnej. Nagroda, aby należeć do uczestnika akcji, musi brać udział w jednej z akcji.

Określenie atrybutów i ich dziedzin:

Tabela 19: Atrybuty akcji marketingowej

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
Id akcji	Integer	Tak	Unikatowy numer akcji marketingowej
Nazwa	VarChar(25)	Tak	Nazwa akcji marketingowej
Opis	VarChar(200)	Tak	Opis akcji marketingowej
Data rozpoczęcia	Date	Tak	Data rozpoczęcia akcji marketingowej
Data zakończenia	Date	Tak	Data zakończenia akcji marketingowej
Liczba Uczestników	Integer	Tak	Liczba uczestników akcji marketingowej

Tabela 20: Atrybuty szczegółów akcji marketingowej

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
Id szczegółu	Integer	Tak	Unikatowy numer szczegółu akcji marketingowej
Status Akcji	VarChar(20)	Tak	Status akcji marketingowej
Całkowity Koszt	Decimal(20,6)	Tak	Całkowity koszt akcji marketingowej,
Target wiekowy	VarChar(15)	Nie	Target wiekowy akcji marketingowej
Lokalizacja akcji	VarChar(20)	Tak	Lokalizacja akcji marketingowej
Forma akcji	VarChar(20)	Nie	Forma akcji marketingowej
Rodzaj akcji	VarChar(20)	Tak	Rodzaj akcji marketingowej
Sponsorzy	VarChar(50)	Nie	Sponsorzy biorący udział w akcji marketingowej

Tabela 21: Atrybuty kanałów akcji marketingowej

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
Id kanału	Integer	Tak	Unikatowy numer kanału akcji marketingowej
Typ kanału	VarChar(20)	Tak	Typ kanału akcji marketingowej
Nazwa Kanału	VarChar(20)	Nie	Nazwa kanału
Opis kanału	VarChar(150)	Tak	Opis kanału akcji marketingowej
Forma kanału	VarChar(30)	Tak	Forma kanału akcji marketingowej
Data utworzenia	Date	Tak	Data utworzenia kanału akcji marketingowej
Liczba odbiorców	Integer	Tak	Liczba odbiorców kanału akcji marketingowej

Tabela 22: Atrybuty uczestników akcji marketingowej

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
Id uczestnika	Integer	Tak	Unikatowy numer uczestnika akcji marketingowej
Imię uczestnika	VarChar(20)	Tak	Imię uczestnika akcji marketingowej
Nazwisko uczestnika	VarChar(30)	Tak	Nazwisko uczestnika akcji marketingowej
Email uczestnika	VarChar(50)	Tak	Email uczestnika akcji marketingowej
Liczba nagród zdobytych	Integer	Tak	Liczba nagród zdobytych przez uczestnika akcji marketingowej

Tabela 23: Atrybuty nagród

Atrybut	Typ i dziedzina	Czy obowiązkowy?	Opis
Id nagrody	Integer	Tak	Unikatowy numer nagrody
Typ nagrody	VarChar(30)	Nie	Typ nagrody
Nazwa nagrody	VarChar(30)	Tak	Nazwa nagrody
Opis nagrody	VarChar(100)	Tak	Opis nagrody

### 3.3. Klucze kandydujące i główne

Nazwa encji	Klucz główny	Klucz kandydujący
AkcjeMarketingowe	idAkcji	nazwa, opis
AkcjeSzczegoly	idSzczegolu	-
KanalyAkcji	idKanal	nazwa, forma
UczestnicyAkcji	idUczestnika	email
Nagrody	idNagrody	-

Tabela 24: Struktura encji, klucze główne i kandydujące

### 3.4. Postać normalna

#### Postać Normalna - Kryterium pierwszej postaci normalnej:

Aby doprowadzić bazę danych do pierwszej postaci normalnej, należy wykonać następujące reguły:

- Każda wartość atrybutu w każdej krotce relacji jest wartością elementarną (atomową), czyli taką, której się nie da rozłożyć.
- Brak powtarzających się grup.

#### Postać Normalna - Kryterium drugiej postaci normalnej:

Kryterium drugiej postaci normalnej zakłada, że pierwsza postać normalna została spełniona oraz wszystkie atrybuty w tabeli, które nie są kluczami, są w pełni funkcyjne i zależne od wszystkich innych kluczy w relacji. Klucz powinien być prosty. W projekcie każdy klucz jest kluczem prostym - identyfikatorem ID.

#### Postać Normalna - Kryterium trzeciej postaci normalnej:

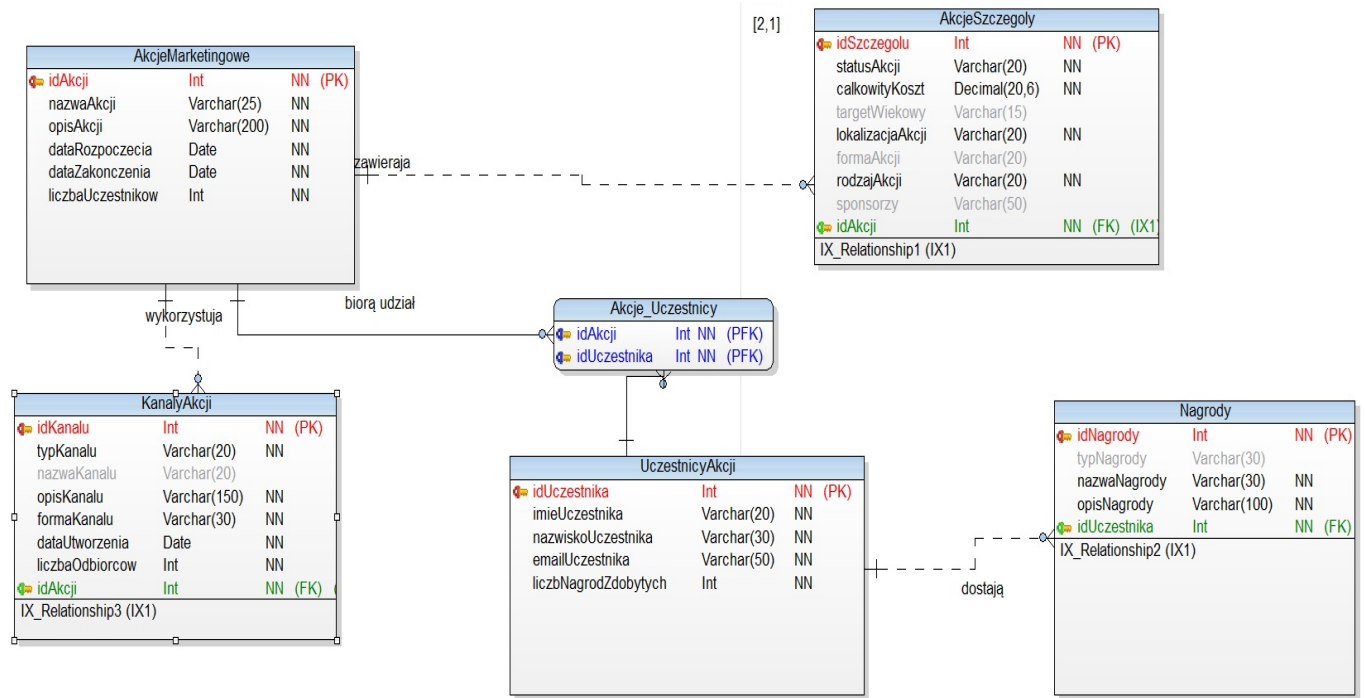
Kryterium trzeciej postaci normalnej zakłada, że druga postać normalna została spełniona oraz żaden atrybut kluczowy nie jest zależny funkcyjnie od innych atrybutów niekluczowych.

#### Więzy integralności:

Więzy integralności jest to zgodność pomiędzy zawartością pól rekordów oraz typami, jakie mogą przyjmować. Dla przykładu jeśli dla atrybutu "nazwaAkcji" jego typ został określony jako varchar(25) to pole to powinno przechowywać taką właśnie wartość a nie na przykład Integer.

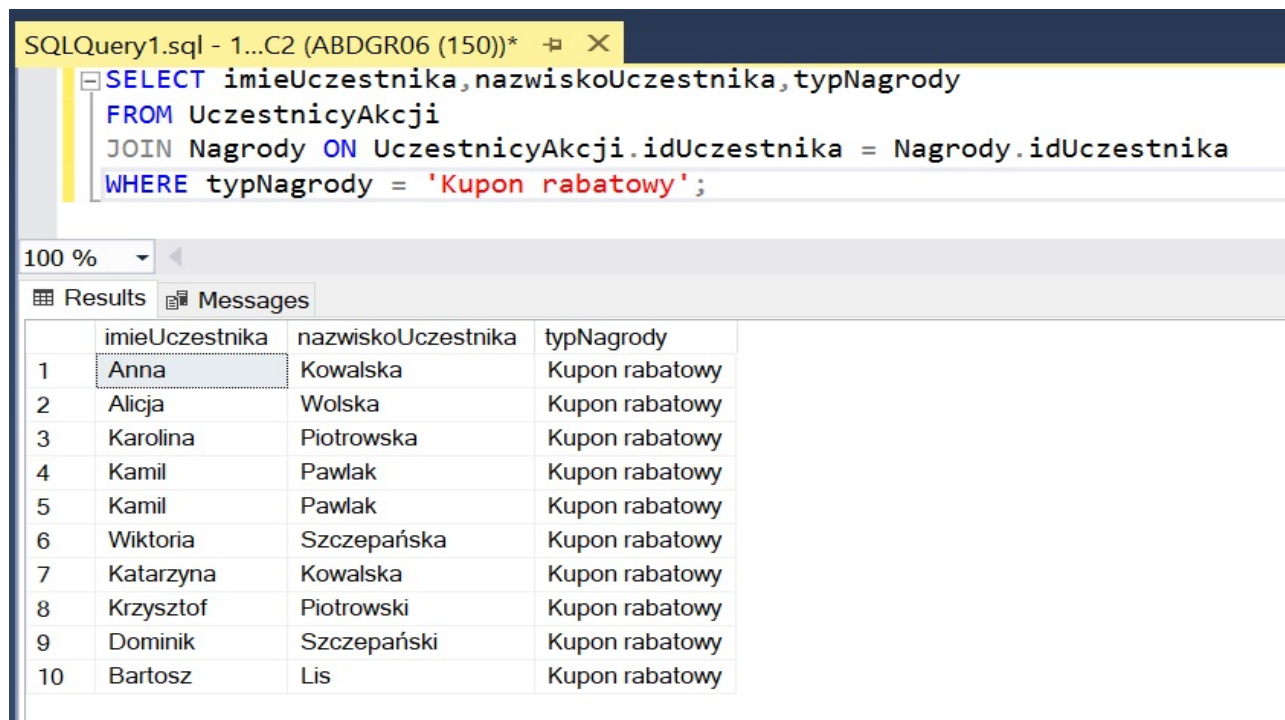
Aby spełnić warunki więzów integralności każde pole jest polem atomowym (co zostało dokonane podczas 1 postaci normalnej), każda encja ma jeden unikatowy klucz główny a problem, że dany atrybut przyjmuje wartość NULL zdarzy się niezwykle rzadko gdyż większość pól jest obowiązkowa.

### 3.5. Schemat modelu fizycznego



Rysunek 7: Schemat modelu fizycznego

### 3.6. Przykłady zapytań i poleceń SQL odnoszących się do bazy danych



The screenshot shows a SQL query editor window titled "SQLQuery1.sql - 1...C2 (ABDGR06 (150))\*". The query is as follows:

```
SELECT imieUczestnika,nazwiskoUczestnika,typNagrody
FROM UczestnicyAkcji
JOIN Nagrody ON UczestnicyAkcji.idUczestnika = Nagrody.idUczestnika
WHERE typNagrody = 'Kupon rabatowy';
```

Below the query editor, the "Results" tab is active, displaying a table with 10 rows of data. The table has three columns: "imieUczestnika", "nazwiskoUczestnika", and "typNagrody".

	imieUczestnika	nazwiskoUczestnika	typNagrody
1	Anna	Kowalska	Kupon rabatowy
2	Alicja	Wolska	Kupon rabatowy
3	Karolina	Piotrowska	Kupon rabatowy
4	Kamil	Pawlak	Kupon rabatowy
5	Kamil	Pawlak	Kupon rabatowy
6	Wiktoria	Szczepańska	Kupon rabatowy
7	Katarzyna	Kowalska	Kupon rabatowy
8	Krzysztof	Piotrowski	Kupon rabatowy
9	Dominik	Szczepański	Kupon rabatowy
10	Bartosz	Lis	Kupon rabatowy

Rysunek 8: Zapytanie którzy uczestnicy otrzymali jako nagrodę kupon rabatowy

WIN-MTDUB73ELLS.A....Akcje\_Uczestnicy SQLQuery2.sql - 1...RC2 (ABDGR06 (62))\* SQLQuery1.sql - 1...RC2 (ABDGR06 (52))\*

```
SELECT typKanal, AVG(liczbaOdbiorcow) AS sredniaLiczbaOdbiorcow
FROM KanalyAkcji
GROUP BY typKanal;
```

100 %

Results Messages

	typKanal	sredniaLiczbaOdbiorcow
1	Bezpieczeństwa	40
2	Edukacyjny	50
3	Informacyjny	66
4	Interakcji z marką	50
5	Interaktywny	100
6	Reklamowy	72
7	Rozrywkowy	67
8	Społecznościowy	150
9	Usługowy	120

Rysunek 9: Średnia liczba odbiorców dla każdego typu kanału akcji

WIN-MTDUB73ELLS.A....Akcje\_Uczestnicy SQLQuery2.sql - 1...RC2 (ABDGR06 (62))\* SQLQuery1.sql - 1...RC2 (ABDGR06 (52))\*

```
SELECT dataRozpoczecia, nazwaAkcji, COUNT(AU.idUczestnika) AS liczbaUczestnikow
FROM AkcjeMarketingowe AM
LEFT JOIN Akcje_Uczestnicy AU ON AM.idAkcji = AU.idAkcji
GROUP BY dataRozpoczecia, nazwaAkcji
ORDER BY dataRozpoczecia;
```

100 %

Results Messages

	dataRozpoczecia	nazwaAkcji	liczbaUczestnikow
1	2023-04-01	SuperPromo2024	40
2	2023-06-01	LetniaFiesta	30
3	2023-07-10	RodzinnaZabawa	12
4	2023-08-20	EkstremalnaPrzygoda	25
5	2023-09-12	Podzielsieswojapinią	0
6	2023-09-15	MagicznyWeekend	30
7	2023-10-05	ParkOdkrywców	21
8	2023-10-25	HalloweenHorror	19
9	2023-12-01	ZimaNałyzwy	35
10	2024-02-10	ZwariowanaKaruzela	37
11	2024-02-11	PiknikNaTrawie	39
12	2024-02-18	PirackiRejs	31
13	2024-02-26	WakacjeWesołychM...	31
14	2024-03-01	FestiwałKolorów	32

Rysunek 10: Policz liczbę uczestników dla każdej akcji marketingowej

### 3.7. Skrypty w SQL

```
/*
Created: 29.02.2024
Modified: 09.03.2024
Model: Microsoft SQL Server 2019
Database: MS SQL Server 2019
*/

-- Create defaults section -----

CREATE DEFAULT Default1 AS
go

-- Create tables section -----

-- Table AkcjeMarketingowe

CREATE TABLE AkcjeMarketingowe
(
    idAkcji Int NOT NULL,
    nazwaAkcji Varchar(25) NOT NULL,
    opisAkcji Varchar(200) NOT NULL,
    dataRozpoczecia Date NOT NULL,
    dataZakonczenia Date NOT NULL,
    liczbaUczestnikow Int NOT NULL
)
go

-- Add keys for table AkcjeMarketingowe

ALTER TABLE AkcjeMarketingowe ADD CONSTRAINT PK_AkcjeMarketingowe PRIMARY KEY (idAkcji)
go

-- Table AkcjeSzczegoly

CREATE TABLE AkcjeSzczegoly
(
    idSzczegolu Int NOT NULL,
    statusAkcji Varchar(20) NOT NULL,
    calkowityKoszt Decimal(20,6) NOT NULL,
    targetWiekowy Varchar(15) NULL,
    lokalizacjaAkcji Varchar(20) NOT NULL,
    formaAkcji Varchar(20) NULL,
    rodzajAkcji Varchar(20) NOT NULL,
    sponsorzy Varchar(50) NULL,
    idAkcji Int NOT NULL
)
go

-- Create indexes for table AkcjeSzczegoly

CREATE INDEX IX_Relationship1 ON AkcjeSzczegoly (idAkcji)
go

-- Add keys for table AkcjeSzczegoly

ALTER TABLE AkcjeSzczegoly ADD CONSTRAINT PK_AkcjeSzczegoly PRIMARY KEY (idSzczegolu)
go
```



```

-- Table KanalyAkcji

CREATE TABLE KanalyAkcji
(
    idKanalů Int NOT NULL,
    typKanalů Varchar(20) NOT NULL,
    nazwaKanalů Varchar(20) NULL,
    opisKanalů Varchar(150) DEFAULT 150 NOT NULL,
    adresKanalů Varchar(30) NOT NULL,
    dataUtworzenia Date NOT NULL,
    liczbaOdbiorców Int NOT NULL,
    idAkcji Int NOT NULL
)
go

-- Create indexes for table KanalyAkcji

CREATE INDEX IX_Relationship3 ON KanalyAkcji (idAkcji)
go

-- Add keys for table KanalyAkcji

ALTER TABLE KanalyAkcji ADD CONSTRAINT PK_KanalyAkcji PRIMARY KEY (idKanalů)
go

-- Table UczestnicyAkcji

CREATE TABLE UczestnicyAkcji
(
    idUczestnika Int NOT NULL,
    imieUczestnika Varchar(20) NOT NULL,
    nazwiskoUczestnika Varchar(30) DEFAULT 30 NOT NULL,
    emailUczestnika Varchar(50) NOT NULL,
    liczbaNagrodyZdobytych Int NOT NULL
)
go

-- Add keys for table UczestnicyAkcji

ALTER TABLE UczestnicyAkcji ADD CONSTRAINT
PK_UczestnicyAkcji PRIMARY KEY (idUczestnika)
go

-- Table Nagrody

CREATE TABLE Nagrody
(
    idNagrody Int NOT NULL,
    typNagrody Varchar(30) NULL,
    nazwaNagrody Varchar(30) NOT NULL,
    opisNagrody Varchar(100) NOT NULL,
    idUczestnika Int NOT NULL
)
go

-- Create indexes for table Nagrody

CREATE INDEX IX_Relationship2 ON Nagrody (idUczestnika)
go

```

```

-- Add keys for table Nagrody

ALTER TABLE Nagrody ADD CONSTRAINT PK_Nagrody PRIMARY KEY (idNagrody)
go

-- Table Akcje_Uczestnicy

CREATE TABLE Akcje_Uczestnicy
(
    idAkcji Int NOT NULL,
    idUczestnika Int NOT NULL
)
go

-- Add keys for table Akcje_Uczestnicy

ALTER TABLE Akcje_Uczestnicy ADD CONSTRAINT PK_Akcje_Uczestnicy
PRIMARY KEY (idAkcji,idUczestnika)
go

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

ALTER TABLE AkcjeSzczegoly ADD CONSTRAINT zawieraja
FOREIGN KEY (idAkcji)
REFERENCES AkcjeMarketingowe (idAkcji) ON UPDATE NO ACTION
ON DELETE NO ACTION
go

ALTER TABLE Nagrody ADD CONSTRAINT dostaja
FOREIGN KEY (idUczestnika)
REFERENCES UczestnicyAkcji (idUczestnika)
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE KanalyAkcji ADD CONSTRAINT wykorzystuja
FOREIGN KEY (idAkcji)
REFERENCES AkcjeMarketingowe (idAkcji) ON UPDATE
NO ACTION ON DELETE NO ACTION
go

ALTER TABLE Akcje_Uczestnicy ADD CONSTRAINT biora udzial FOREIGN KEY (idAkcji)
REFERENCES AkcjeMarketingowe (idAkcji)
ON UPDATE NO ACTION ON DELETE NO ACTION
go

ALTER TABLE Akcje_Uczestnicy ADD CONSTRAINT Relationship2 FOREIGN KEY (idUczestnika)
REFERENCES UczestnicyAkcji (idUczestnika)
ON UPDATE NO ACTION ON DELETE NO ACTION
go

```