

Piotr Majecki 293242
Mateusz Łaguna 293232
Prowadzący: dr inż. Marcin Kowalczyk

Projekt z przedmiotu Wprowadzenie Do Baz Danych (WBD) część I

„Park Rozrywki”

Spis treści:

1.ZAKRES I CEL PROJEKTU:	3
2.DEFINICJA SYSTEMU:	3
2.1 PERSPEKTYWY UŻYTKOWNIKÓW:	4
3. MODEL KONCEPTUALNY:	6
3.1 DEFINICJA ZBIORÓW ENCJI OKREŚLONYCH W PROJEKCIE:	6
3.2 USTALENIE ZWIĄZKÓW MIĘDZY ENCJAMI I ICH TYPÓW:	7
3.3 OKREŚLENIE ATRYBUTÓW I ICH DZIEDZIN:	9
3.4 DODATKOWE REGUŁY INTEGRALNOŚCIOWE:	14
3.5 KLUCZE KANDYDUJĄCE I GŁÓWNE:	15
3.6 SCHEMAT ER NA POZIOMIE KONCEPTUALNYM:	16
3.7 PROBLEM PUŁAPEK SZCZELINOWYCH I WACHLARZOWYCH:	16
4. MODEL LOGICZNY:	18
4.1 CHARAKTERYSTYKA MODELU RELACYJNEGO:	18
4.2 USUNIĘCIE WŁAŚCIWOŚCI NIEKOMPATYBILNYCH Z MODELEM RELACYJNYM:	19
4.3 PROCES NORMALIZACJI:	25
4.4 SCHEMAT ER NA POZIOMIE MODELU LOGICZNEGO:	26
4.5 WIĘZY INTEGRALNOŚCI:	27
4.6 PROCES DENORMALIZACJI:	27
5. FAZA FIZYCZNA:	29
5.1 PROJEKT TRANSAKCJI I WERYFIKACJA ICH WYKONALNOŚCI:	29
5.2 STROJENIE BAZY DANYCH – DOBÓR INDEKSÓW:	30
5.3 SKRYPT SQL ZAKŁADAJĄCY BAZĘ DANYCH:	32
5.4 PRZYKŁADY ZAPYTAŃ I POLECEŃ SQL ODNOSZĄCYCH SIĘ DO BAZY DANYCH:	39
6. BIBLIOGRAFIA	41

1. Zakres i cel projektu:

- Cel projektu:

Celem projektu było zapoznanie się z projektowaniem i tworzeniem baz danych. Należało zaprojektować relacyjną bazę danych na poziomie koncepcyjnym, logicznym oraz fizycznym.

Projekt oparty jest o system zarządzania bazami danych firmy Oracle i jest napisany w języku SQL.

Oprogramowanie, którego użyliśmy do wykonania bazy danych to:

1. Toad Data Modeler 7.1
2. Oracle Database 19c
3. SQLdeveloper (version 19.2.1)

- Zakres projektu:

Realizowany projekt umożliwia obsługę i zarządzanie parkiem rozrywki. System, który stworzyliśmy pozwala na przetrzymywanie danych na temat parku rozrywki (jego nazwę, adres, powierzchnię, liczbę atrakcji, itp.), atrakcji i udogodnień jakie się w nim znajdują, z podziałem na kategorie dla jakich osób są one przeznaczone oraz zawiera dane na temat pracowników (imię, nazwisko, adres, datę urodzenia, numer konta bankowego czy dokumentu, itp.), z wyróżnieniem ich wynagrodzenia czy stanowiska, jak i również dane na temat Naszych klientów (imię, nazwisko, wzrost, itp.). Nasza baza danych przechowuje również dane związane z rodzajami biletów, wagonikami do różnych atrakcji oraz kasami, od kiedy i do kiedy są czynne.

Należy tu zaznaczyć, że Nasz park rozrywki ma jedynie jedną placówkę w kraju, co oznacza, że ma tylko jednego właściciela.

2. Definicja systemu:

- Funkcjonalność systemu:

1. Możliwość modyfikowania danych związanych z parkiem rozrywki - nazywanie parku rozrywki, określenie jego powierzchni, liczby atrakcji, przechowywanie jego adresu, numeru telefonu, adresu mailowego oraz całej wiedzy na temat właściciela obiektu.
2. Wgląd w dane związane z parkiem rozrywki.
3. Dodawanie oraz usuwanie pracowników.
4. Modyfikowanie danych na temat pracowników, tj. ich imiona, nazwiska, daty urodzenia, daty zatrudnienia, adresy zamieszkania, numery telefonu. W Naszej bazie przetrzymywane są również dane związane z numerami kont bankowych oraz numerami dokumentów tożsamości. Opcjonalnie pracownik może podać również swoją płeć, PESEL oraz adres e-mail.

5. Wgląd do danych związanych z pracownikami zatrudnionymi w Naszym parku rozrywki.
6. Określanie wynagrodzenia (+premie) dla pracowników oraz datę ich wypłacania.
7. Możliwość modyfikacji danych związanych z wynagrodzeniem.
8. Wgląd w dane związane z wynagrodzeniem pracownika.
9. Przeglądanie danych osobowych klientów, którzy byli w Naszym parku rozrywki, tj. ich imiona oraz nazwiska. Opcjonalnie możemy również posiadać dane na temat: płci, daty urodzenia, wzrostu, adresu mailowego.
10. Wprowadzanie, usuwanie oraz modyfikowanie danych odnośnie klientów.
11. Podgląd kas jakie są otwarte (ich numery) oraz godziny ich pracy w parku rozrywki.
12. Modyfikowanie danych związanych z kasami.
13. Wyświetlanie danych na temat atrakcji znajdujących się w naszym parku rozrywki (nazwa, rodzaj, czas trwania atrakcji oraz ilość wagoników, które się w niej znajdują) z możliwością podziału na atrakcje dla dorosłych, młodzieży i dzieci (w których różnicą może być wiek lub wzrost).
14. Określanie straszności atrakcji w skali: wcale, trochę, straszna, bardzo straszna.
15. Określenie wagoników jakie są potrzebne do danej atrakcji (Mamy tu do wyboru wagoniki: szynowe, latające, pływające, czy gokarty).
16. Określenie marki i modelu wagoników.
17. Modyfikacja danych związanych z atrakcjami oraz wagonikami danej atrakcji.
18. Pokazanie danych na temat przekąsek jakie są sprzedawane w parku rozrywki – ich nazwa, cena oraz skład.
19. Modyfikowanie, dodawanie oraz usuwanie danych związanych z przekąskami, które kupują Nasi klienci.
20. Wgląd do danych związanych z biletami jakie są możliwe do kupienia w kasach naszego parku rozrywki. Pokazanie ich rodzaju (występują bilety jednodniowe oraz trzydniowe), ich ceny oraz daty sprzedaży danego biletu.
21. Możliwość modyfikowania danych o biletach.
22. Możliwość określania rodzajów biletów.
23. Modyfikowanie struktury bazy danych.

2.1 Perspektywy użytkowników:

- **Administrator:**
Jest to osoba, która ma pełny wgląd do systemu i struktur bazy danych. Posiada uprawnienia administratora bazy danych Oracle (tworzony podczas instalacji).
- **Właściciel:**
Osoba upoważniona do wglądu i określania wszystkich danych znajdujących się w bazie danych, lecz nie powinna (a nawet nie może i jest to zrobione celowo) ona ingerować w ich strukturę oraz ich modyfikować (modyfikację powinna zostawić osobom, które są zatrudnione do tego celu – czyli dział księgowości/kadr). Właściciel przekazuje swoje uwagi co do modyfikacji w bazie danych swoim pracownikom i z jego rozkazu dane są modyfikowane, ale sam ich nie modyfikuje.

- **Księgowość/kadry:**
Dział, w którym znajdują się osoby mające możliwość modyfikowania danych związanych z parkiem rozrywki, pracownikami jakich zatrudnia placówka oraz klientami, którzy odwiedzają park rozrywki. Dana osoba ma również wgląd do wszystkich danych, jednak nie może ich określać.
- **Pracownik:**
Osoba mogąca wnikać i modyfikować dane w bazie danych związanych z własną osobą oraz przeglądać dane związane z klientami i parkiem rozrywki.
- **Klient:**
Ma możliwość wglądu i modyfikacji swoich danych osobowych przechowywanych w bazie danych oraz może przeglądać oferty naszego parku rozrywki oraz jego zasoby.

Funkcjonalność	Administrator	Właściciel	Księgowość/kadry	Pracownik	Klient
1	Tak	Nie	Tak	Nie	Nie
2	Tak	Tak	Tak	Tak	Tak
3	Tak	Tak	Tak	Nie	Nie
4	Tak	Nie	Tak	Tylko siebie	Nie
5	Tak	Tak	Tak	Tak	Nie
6	Tak	Tak	Nie	Nie	Nie
7	Tak	Nie	Tak	Nie	Nie
8	Tak	Tak	Tak	Tak	Nie
9	Tak	Tak	Tak	Tak	Tak
10	Tak	Nie	Tak	Nie	Tylko siebie
11	Tak	Tak	Tak	Tak	Tak
12	Tak	Nie	Tak	Nie	Nie
13	Tak	Tak	Tak	Tak	Tak
14	Tak	Tak	Nie	Nie	Nie
15	Tak	Tak	Nie	Nie	Nie
16	Tak	Tak	Nie	Nie	Nie
17	Tak	Nie	Tak	Nie	Nie
18	Tak	Tak	Tak	Tak	Tak
19	Tak	Nie	Tak	Nie	Nie
20	Tak	Tak	Tak	Tak	Tak
21	Tak	Nie	Tak	Nie	Nie
22	Tak	Tak	Nie	Nie	Nie
23	Tak	Nie	Nie	Nie	Nie

3. Model konceptualny:

3.1 Definicja zbiorów encji określonych w projekcie:

- Encja park rozrywki – istnieje tylko jeden obiekt tej encji. Atrybuty tej encji określają Nasz park rozrywki.
- Encja pracownik – istnieje wiele obiektów tej encji. Reprezentuje ona pracowników zatrudnionych w parku rozrywki na stanowisku: kasa lub atrakcja. Przechowuje atrybuty opisujące pracownika.
- Encja wynagrodzenie – istnieje wiele obiektów tej encji. Jej atrybuty opisują wynagrodzenie jakie dostaje pracownik, pracujący w parku rozrywki. Może uwzględniać dodatki do wynagrodzenia (premie).
- Encja klient – istnieje wiele obiektów tej encji. Zawiera atrybuty opisujące dane personalne klientów parku rozrywki.
- Encja atrakcja – występuje wiele obiektów tej encji. Atrybuty znajdujące się w tej encji określają atrakcje parku rozrywki. Wyróżniamy kilka rodzajów atrakcji: dla dorosłych, dla dzieci oraz dla młodzieży.
- Encja wagonik – występuje wiele obiektów tej encji. Jej atrybuty określają wagoniki jakich używa się w danej atrakcji. Istnieje kilka rodzajów wagoników, a są to: szynowy, latający, pływający, gokart.
- Encja Dla_doroslych (child entity of Atrakcja) – encja pochodna od encji Atrakcja (dziedziczenie) – określa atrakcje dla dorosłych. Występuje wiele obiektów tej encji, a jej atrybuty ściślej określają dany rodzaj atrakcji. Są rozszerzone o wymagany wzrost, wymagany wiek, straszność atrakcji oraz czy klient posiada choroby serca.
- Encja Dla_dzieci (child entity of Atrakcja) - encja pochodna od encji Atrakcja (dziedziczenie) – określa atrakcje dla dzieci. Występuje wiele obiektów tej encji, a jej atrybuty ściślej określają dany rodzaj atrakcji. Są rozszerzone o atrybuty nadzór rodzica (który określa czy dziecko musi być pod opieką rodzica na danej atrakcji) oraz o ściślejsze określenie dla jakich dzieci jest ta atrakcja, tj. niemowlaka, dziecka, dużego dziecka.
- Encja dla_mlodziezy (child entity of Atrakcja) - encja pochodna od encji Atrakcja (dziedziczenie) – określa atrakcje dla młodzieży. Występuje wiele obiektów tej encji (w przykładowej bazie danych występuje akurat tylko jedna taka atrakcja), a jej atrybuty ściślej określają dany rodzaj atrakcji. Są rozszerzone o wymagany wiek, wymagany wzrost oraz określają straszność atrakcji.
- Encja bilet – występuje wiele obiektów tej encji. Atrybuty tej encji opisują bilety, które są wymagane, aby skorzystać z danej atrakcji. Bilety kupuje się w kasach parku rozrywki. Wyróżniamy dwa rodzaje biletów: jednodniowe oraz trzydniowe.
- Encja kasa – występuje wiele obiektów tej encji. Atrybuty, które się w niej znajdują opisują kasy parku rozrywki. W kasach są sprzedawane bilety, które musi zakupić klient, aby skorzystać z danej atrakcji (bez niego nie da się skorzystać z atrakcji).
- Encja przekaska – występuje wiele obiektów tej encji. Jej atrybuty opisują przekąski, jakie są sprzedawane w Naszym parku rozrywki.

3.2 Ustalenie związków między encjami i ich typów:

Wszystkie automatycznie wygenerowane tabelki, które znajdują w całym pliku są zrobione na podstawie specyfikacji modelu conceptualnego lub logicznego wykonanego w TOAD Data Modelerze.

Wszystkie związki pomiędzy encjami są binarne.

- Związki encji park rozrywki:

Park rozrywki – Pracownik: obowiązkowy – opcjonalny

Park rozrywki – Atrakcja: obowiązkowy – opcjonalny

Park rozrywki – Klient: obowiązkowy – opcjonalny

Park rozrywki – Kasa: obowiązkowy – opcjonalny

Park rozrywki – Przekaska: obowiązkowy – opcjonalny

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Zatrudnia	Non-Identifying	Park_rozrywki - Pracownik	1..1 - 0..m
Zapewnia	Non-Identifying	Park_rozrywki - Atrakcja	1..1 - 0..m
Zabawia	Non-Identifying	Park_rozrywki - Klient	1..1 - 0..m
Ma	Non-Identifying	Park_rozrywki - Kasa	1..1 - 0..m
Umożliwia	Non-Identifying	Park_rozrywki - Przekaska	1..1 - 0..m

- Związki encji pracownik:

Pracownik – Park rozrywki: opcjonalny – obowiązkowy

Pracownik – Wynagrodzenie: obowiązkowy – opcjonalny

Pracownik – Atrakcja: obowiązkowy – opcjonalny

Pracownik – Kasa: opcjonalny – opcjonalny

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Zatrudnia	Non-Identifying	Park_rozrywki - Pracownik	1..1 - 0..m
Otrzymuje	Non-Identifying	Pracownik - Wynagrodzenie	1..1 - 0..m
Pracuje przy	Non-Identifying	Pracownik - Atrakcja	1..n - 0..m
Obsługuje	Non-Identifying	Pracownik - Kasa	0..n - 0..m

- Związki encji wynagrodzenie:

Wynagrodzenie – Pracownik: opcjonalny – obowiązkowy

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Otrzymuje	Non-Identifying	Pracownik - Wynagrodzenie	1..1 - 0..m

- Związki encji klient:

Klient – Park rozrywki: opcjonalny – obowiązkowy

Klient – Bilet: opcjonalny – opcjonalny

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Zabawia	Non-Identifying	Park rozrywki - Klient	1..1 - 0..m
Kupuje	Non-Identifying	Klient - Bilet	0..1 - 0..m

- Związki encji atrakcja:

Atrakcja – Park rozrywki: opcjonalny – obowiązkowy

Atrakcja – Wagonik: obowiązkowy – opcjonalny

Atrakcja – Pracownik: opcjonalny – obowiązkowy

Atrakcja – Bilet: obowiązkowy – opcjonalny

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Zapewnia	Non-Identifying	Park rozrywki - Atrakcja	1..1 - 0..m
Posiada	Non-Identifying	Atrakcja - Wagonik	1..1 - 0..m
Pracuje przy	Non-Identifying	Pracownik - Atrakcja	1..n - 0..m
Wymaga	Non-Identifying	Atrakcja - Bilet	1..n - 0..m

- Związki encji wagonik:

Wagonik – Atrakcja: opcjonalny – obowiązkowy

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Posiada	Non-Identifying	Atrakcja - Wagonik	1..1 - 0..m

- Związki encji bilet:

Bilet – Klient: opcjonalny – opcjonalny

Bilet – Kasa: opcjonalny – obowiązkowy

Bilet – Atrakcja: opcjonalny – obowiązkowy

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Kupuje	Non-Identifying	Klient - Bilet	0..1 - 0..m
Sprzedaje	Non-Identifying	Kasa - Bilet	1..1 - 0..m
Wymaga	Non-Identifying	Atrakcja - Bilet	1..n - 0..m

- Związki encji kasa:

Kasa – Park rozrywki: opcjonalny – obowiązkowy

Kasa – Bilet: obowiązkowy – opcjonalny

Kasa – Pracownik: opcjonalny – opcjonalny

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Ma	Non-Identifying	Park rozrywki - Kasa	1..1 - 0..m
Sprzedaje	Non-Identifying	Kasa - Bilet	1..1 - 0..m
Obsługuje	Non-Identifying	Pracownik - Kasa	0..n - 0..m

- Związki encji przekąski:

Przekaska – park rozrywki: opcjonalny – obowiązkowy

Relationships

Relationship Name	Relationship Type	Relationship Between	Cardinality
Umożliwia	Non-Identifying	Park rozrywki - Przekaska	1..1 - 0..m

3.3 Określenie atrybutów i ich dziedzin:

Wyjaśnienie:

M¹⁾ - Mandatory

D²⁾ - Default

R³⁾ - Rule

- Encja park rozrywki:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_parku_rozrywki		Integer	YES	NO	NO	Unikatowy numer parku rozrywki
NO	Nazwa_parku		VarChar(30)	YES	NO	NO	Nazwa parku rozrywki
NO	Adres_parku		VarChar(400)	YES	NO	NO	Adres parku rozrywki
NO	Powierzchnia		Integer	YES	NO	NO	Powierzchnia parku rozrywki
NO	Ilosc_atrakcji		Integer	YES	NO	NO	Ilosc atrakcji w parku rozrywki
NO	Numer_telefonu_parku		VarChar(9)	YES	NO	NO	Numer telefonu do parku rozrywki
NO	Adres_mailowy_parku		VarChar(60)	YES	NO	NO	Adres e-mail do parku rozrywki
NO	Wlasciciel		VarChar(60)	YES	NO	NO	Wlasciciel parku rozrywki

- Encja pracownik:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_pracownika		Integer	YES	NO	NO	Unikatowy numer pracownika
NO	Imie_pracownika		VarChar(20)	YES	NO	NO	Imie pracownika
NO	Nazwisko_pracownika		VarChar(30)	YES	NO	NO	Nazwisko pracownika
NO	Plec_pracownika	Plec	Character(1)	NO	NO	YES	Plec pracownika
NO	Data_urodzenia_pracownika		Date	YES	NO	NO	Data urodzenia pracownika
NO	PESEL		Character(11)	NO	NO	NO	PESEL pracownika
NO	Stanowisko	Stanowisko	VarChar(20)	YES	NO	YES	Stanowiski na jakim pracuje pracownik
NO	Data_zatrudnienia		Date	YES	NO	NO	Data zatrudnienia pracownika
NO	Numer_dokument		Character(9)	YES	NO	NO	Numer dokumentu pracownika
NO	Numer_konta		Character(26)	YES	NO	NO	Numer konta bankowego pracownika
NO	Adres_pracownika		VarChar(400)	YES	NO	NO	Adres zamieszkania pracownika
NO	Numer_telefonu_pracownika		VarChar(9)	YES	NO	NO	Numer telefonu pracownika
NO	Adres_mailowy_pracownika		VarChar(60)	NO	NO	NO	Adres e-mail pracownika

Domain Properties
Plec
General Rules To Do
Caption: Plec, Name: Plec
Data Type: Character(x), Default Rule: No default
Length: 1
OK Ok+Add Cancel Apply

Rule Properties
Plec
General SQL To Do
1 Plec IN ('K','M');
OK Ok+Add Cancel Apply

Domain Properties
Stanowisko
General Rules To Do
Caption: Stanowisko, Name: Stanowisko
Data Type: Variable Character(x), Default Rule: No default
Length: 20
OK Ok+Add Cancel Apply

Rule Properties
Stanowisko
General SQL To Do
1 Stanowisko IN ('kasa','atrakcja');
OK Ok+Add Cancel Apply

- Encja wynagrodzenie:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_wynagrodzenia		Integer	YES	NO	NO	Unikatowy numer wynagrodzenia
NO	Data_wypłacania		Date	YES	NO	NO	Data wypłacania wynagrodzenia
NO	Wysokosc		Money	YES	NO	NO	Wysokosc wynagrodzenia
NO	Dodatki		Money	NO	NO	NO	Dodatki do wynagrodzenia

- Encja klient:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_klienta		Integer	YES	NO	NO	Unikatowy numer klienta
NO	Imie_klienta		VarChar(20)	YES	NO	NO	Imie klienta
NO	Nazwisko_klienta		VarChar(30)	YES	NO	NO	Nazwisko klienta
NO	Plec_klienta	Plec	Character(1)	NO	NO	YES	Plec klienta
NO	Data_urodzenia_klienta		Date	NO	NO	NO	Data urodzenia klienta
NO	Wzrost		Integer	NO	NO	NO	Wzrost klienta
NO	Adres_mailowy_klienta		VarChar(60)	NO	NO	NO	Adres e-mail klienta

Domena i reguła odnosząca się do Plec znajduje się wyżej (encja pracownik).

- Encja atrakcja:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_atrakcji		Integer	YES	NO	NO	Unikatowy numer atrakcji
NO	Nazwa_atrakcji		VarChar(30)	YES	NO	NO	Nazwa atrakcji
NO	Rodzaj_atrakcji	Rodzaj_atrakcji	VarChar(20)	YES	NO	YES	Rodzaj atrakcji
NO	Ilosc_wagonikow		Integer	YES	NO	NO	Ilosc wagonikow w danej atrakcji
NO	Czas_trwania		Integer	YES	NO	NO	Czas trwania atrakcji

Domain Properties

Rodzaj_atrakcji

General Rules To Do

Caption: Rodzaj atrakcji

Name: Rodzaj_atrakcji

Data Type: Variable Character(x)

Default Rule: No default

Length: 20

OK Ok+Add Cancel Apply

Rule Properties

Rodzaj_atrakcji

General SQL To Do

i Rodzaj_atrakcji IN ('dla_dzieci','dla_mlodziezy','dla_doroslych');

OK Ok+Add Cancel Apply

- Encja wagonik:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_wagonika		Integer	YES	NO	NO	Unikatowy numer wagonika
NO	Marka		VarChar(30)	YES	NO	NO	Marka wagonika
NO	Model		VarChar(30)	YES	NO	NO	Model wagonika
NO	Rodzaj_wagonika	Rodzaj_wagonika	VarChar(20)	YES	NO	YES	Rodzaj wagonika w danej atrakcji

Domain Properties

Rodzaj_wagonika

General Rules To Do

Caption: Rodzaj wagonika
Name: Rodzaj_wagonika
Data Type: Variable Character(x)
Default Rule: No default
Length: 20

OK Ok+Add Cancel Apply

Rule Properties

Rodzaj_wagonika

General SQL To Do

1 Rodzaj_wagonika IN ('szynowy','latajacy','plywajacy','gokart');

OK Ok+Add Cancel Apply

- Encja dla_doroslych (child entity of Atrakcja):

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
NO	Wymagany_wiek		Integer	YES	NO	NO	Wymagany wiek aby skorzystac z atrakcji
NO	Wymagany_wzrost		Integer	YES	NO	NO	Wymagany wzrost aby skorzystac z atrakcji
NO	Straszosc	Straszosc	VarChar(20)	YES	NO	YES	Sztraszosc atrakcji
NO	Choroby_serca		Boolean	YES	NO	NO	Czy klient posiada choroby serca

Domain Properties

Straszosc

General Rules To Do

Caption: Straszosc
Name: Straszosc
Data Type: Variable Character(x)
Default Rule: No default
Length: 20

OK Ok+Add Cancel Apply

Rule Properties

Straszosc

General SQL To Do

1 Straszosc IN ('wcale','troche','straszna','bardzo_straszna');

OK Ok+Add Cancel Apply

- Encja dla_dzieci (child entity of Atrakcja):

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
NO	Nadzor_rodzica		Boolean	YES	NO	NO	Czy wymagany jest nadzor rodzica
NO	Dla_jakich	Dla_jakich	VarChar(20)	YES	NO	YES	Dla jakich dzieci przeznaczona jest atrakcja

The image shows two windows from a database design tool. The left window, 'Domain Properties', is for the 'Dla_jakich' attribute. It shows the 'Caption' as 'Dla jakich' and the 'Name' as 'Dla_jakich'. The 'Data Type' is set to 'Variable Character(x)' and the 'Length' is 20. The right window, 'Rule Properties', is for the 'Dla_jakich' rule. It shows a single rule with the SQL expression: 'Dla_jakich IN ('niemowlak','dziecko','duze_dziecko');'.

- Encja dla_mlodziezy (child entity of Atrakcja):

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
NO	Wymagany_wiek		Integer	YES	NO	NO	Wymagany wiek aby skorzystac z atrakcji
NO	Wymagany_wzrost		Integer	YES	NO	NO	Wymagany wzrost aby skorzystac z atrakcji
NO	Straszosc	Straszosc	VarChar(20)	YES	NO	YES	Sztraszosc atrakcji

Domena i reguła odnosząca się do Straszosc znajduje się wyżej (encja dla_doroslych).

- Encja kasa:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_kasy		Integer	YES	NO	NO	Unikatowy numer kasy
NO	Numer_kasy		VarChar(2)	YES	NO	NO	Numer kasy
NO	Otwarta		Time	YES	NO	NO	Czas otwarcia kasy
NO	Zamknieta		Time	YES	NO	NO	Czas zamknięcia kasy

- Encja bilet:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_biletu		Integer	YES	NO	NO	Unikatowy numer biletu
NO	Rodzaj_biletu	Rodzaj_biletu	VarChar(20)	YES	NO	YES	Rodzaj biletu
NO	Cena_biletu		Money	YES	NO	NO	Cena biletu
NO	Data		Date	YES	NO	NO	Data sprzedaży biletu

Domain Properties

Rodzaj_biletu

General

Rules

To Do

Caption

Rodzaj biletu

Name

Rodzaj_biletu

Data Type

Variable Character(x)

Default Rule

No default

Length

20

OK

Ok+Add

Cancel

Apply

Rule Properties

Rodzaj_biletu

General

SQL

To Do

1

Rodzaj_biletu IN ('jednodniowy','trzydniowy');

OK

Ok+Add

Cancel

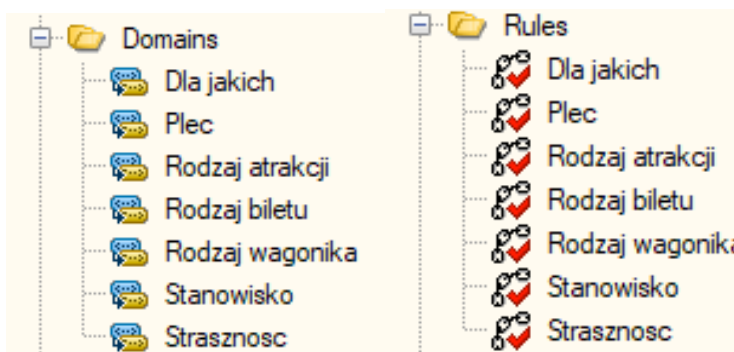
Apply

- Encja przekaska:

Attributes

PUI	Attribute Name	Domain	Data Type	M ¹⁾	D ²⁾	R ³⁾	Description
YES	Id_przekaski		Integer	YES	NO	NO	Unikatowy numer przekaski
NO	Nazwa_przekaski		VarChar(30)	YES	NO	NO	Nazwa przekaski
NO	Cena_przekaski		Money	YES	NO	NO	Cena przekaski
NO	Sklad		VarChar(400)	YES	NO	NO	Opis skladnikow znajdujacych sie w przekasce

3.4 Dodatkowe reguły integralnościowe:



- Dla jakich – reguła występująca w encji dla_dzieci. Określa dla jakich dokładnie dzieci jest dana atrakcja, tj. niemowlaka, dziecka, dużego dziecka.
- Plec – reguła występująca w encji pracownik oraz klient. Przyjmuje dwie wartości tj. 'K' – kobieta lub 'M' – mężczyzna.
- Rodzaj atrakcji – reguła występująca w encji atrakcja. Określa rodzaj atrakcji, a więc dla kogo jest przeznaczona, tj. dla dorosłych, dla dzieci, dla młodzieży.
- Rodzaj biletu – reguła występująca w encji bilet. Określa rodzaj biletu, tj. czy jest on jednodniowy czy trzydniowy.
- Rodzaj wagonika – reguła występująca w encji wagonik. Określa rodzaj wagonika, jaki jest wykorzystywany do danej atrakcji, tj. szynowy, latający, pływający, gokart.
- Stanowisko – reguła występująca w encji pracownik. Określa stanowisko na jakim pracuje dany pracownik, a może on pracować na stanowisku: kasa lub atrakcja.
- Straszność – reguła występująca w encji dla_doroslych oraz dla_mlodzięzy. Określa straszność danej atrakcji, a jest ona mierzona w skali: wcale, trochę, straszna, bardzo straszna.

3.5 Klucze kandydujące i główne:

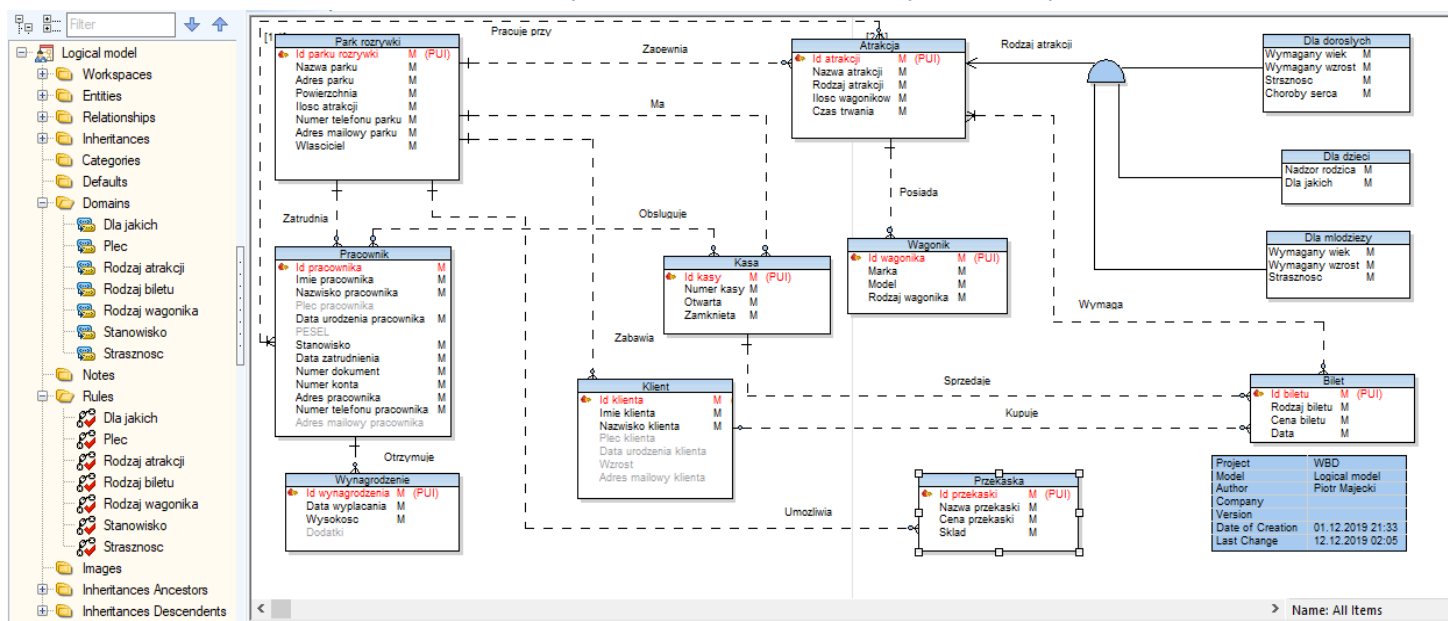
Aby uniknąć niejednoznaczności oraz błędów w Naszej bazie danych postanowiliśmy stworzyć sztuczne klucze główne, które identyfikujemy jako ID danej encji. Dzięki takiemu rozwiązaniu mamy gwarancję, że każdy Nasz klucz główny jest unikatowy, co daje Nam spójność bazy danych.

Jak widzimy niektóre encje nie mają w ogóle odpowiedniego klucza kandydującego. Jeżeli chodzi o encję pracownik mogłoby się wydawać, że PESEL byłby dobrym kluczem głównym, jednak w Naszym parku rozrywki mogą również pracować obcokrajowcy (którzy nie posiadają numeru PESEL), dlatego w encji pracownik ten atrybut nie jest w ogóle wymagany (mandatory) – co z automatu skreśla go z możliwości bycia kluczem głównym. Jeżeli chodzi o numer dokumentu, to ta opcja byłaby o wiele lepsza niż PESEL i mogłaby się sprawdzić w Naszym modelu bazy danych. W wynagrodzeniu ciekawym rozwiązaniem mogłoby być dodatnie jakiegoś odnośnika do numeru konta bankowego, dzięki któremu moglibyśmy zapewnić unikatowość tego atrybutu i ustawić go jako klucz główny tej encji. Dla klienta można by rozważyć podobne rozwiązanie jak dla pracownika, czyli za pomocą numeru dokumentu (należałoby wtedy stworzyć odpowiedni atrybut encji klient). Dla encji wagonik moglibyśmy stworzyć unikatowe numery (każdy wagonik miałby swój, jedyny w swoim rodzaju numer – tak jak np. na wyciągach narciarskich). Jeżeli chodzi o encję bilet to również moglibyśmy stworzyć jakieś unikatowe numery, które same by się generowały podczas wydruku biletu w kasie.

W innych encjach już na tym poziomie udałooby się znaleźć jakieś klucze główne (w głównej mierze odnosiłyby się one do nazwy danej encji).

Encja	Klucz główny	Klucz kandydujący
park rozrywki	Id parku rozrywki	Nazwa parku
pracownik	Id pracownika	Numer dokumentu/PESEL
wynagrodzenie	Id wynagrodzenia	-
klient	Id klienta	-
atrakcja	Id atrakcji	Nazwa atrakcji
wagonik	Id wagonika	-
bilet	Id biletu	-
kasa	Id kasy	Numer kasy
przekaska	Id przekaski	Nazwa przekaski

3.6 Schemat ER na poziomie konceptualnym:



3.7 Problem pułapek szczelinowych i wachlarzowych:

W Naszym modelu bazy danych nie występują ani pułapki wachlarzowe, ani pułapki szczelinowe.

- Jeżeli jednak chcielibyśmy stworzyć pułapkę wachlarzową (co oczywiście jest zabronione, ale w ramach projektu chcemy przeanalizować co należałoby w takiej sytuacji zrobić) to moglibyśmy stworzyć encję placówka (czyli park rozrywki miałby wiele placówek, co oznacza, że istniałoby kilka parków rozrywki). Wtedy, w naszym modelu bazy danych, jeżeli pracownik byłby połączony związkiem z parkiem rozrywki (tak jak dotychczas) to wystąpiłby wachlarz możliwości, czyli nie byłoby wiadomo w której placówce pracownik by tak naprawdę pracował. Rozwiązanie tego problemu polegałoby na zmianie struktury modelu na następujący: park rozrywki posiada placówkę, a to placówka zatrudniałaby danego pracownika.

- Natomiast, jeżeli chodzi o pułapkę szczelinową (oczywiście również zabroniona), to aby ją stworzyć musielibyśmy dodać encję oddział (w danym oddziale pracowaliby pracownicy). Wtedy encja pracownik byłaby połączona związkiem z encją oddział (i nie byłaby połączona z encją park rozrywki). W takiej sytuacji mógłby zaistnieć problem, że jakiś pracownik nie pracuje w danym oddziale, np. nie pracuje w żadnym oddziale (bo pracuje ogólnie w parku rozrywki) albo nie jest ściśle przypisany do jednego oddziału. Należałoby wtedy połączyć encję pracownik z encją park rozrywki (i zostawić wcześniejsze połączenie z oddziałem), co uniemożliwiłoby stworzenie „szczeliny” (braku) w danych.

Dlatego właśnie aby uprościć Nasz model oraz nie wyjść poza ograniczenia projektowe (20 encji) postanowiliśmy nie tworzyć placówek oraz oddziałów, tylko stworzyć jeden park rozrywki oraz przypisać każdego pracownika właśnie do niego.

4. Model logiczny:

4.1 Charakterystyka modelu relacyjnego:

Nasz model logiczny stworzyliśmy w programie Toad Data Modeler 7.1. W celu uzyskania modelu logicznego musieliśmy usunąć niekompatybilności z modelem relacyjnym. Uzyskaliśmy to za pomocą usunięcia związków wiele do wielu (M:N) i zastąpieniu ich tablicami bridge'ującymi. Zmieniliśmy również nazwy encji z liczby pojedynczej na mnogą, dzięki czemu jesteśmy w stanie zachować unikatowość nazw. Każda encja miała ustalony klucz główny jako prosty już na poziomie modelu konceptualnego, dlatego nie musieliśmy niczego zmieniać odnośnie kluczy w modelu logicznym. Kolejnym krokiem, była weryfikacja atrybutów encji, pod kątem pól segmentowych, które nie mogą występować w modelu logicznym (nie spełniają 1PN). Usunęliśmy atrybut właściciel ze zbioru encji parki rozrywki i dodaliśmy nowy zbiór encji o nazwie Wlasciciele, które połączyliśmy ze sobą związkiem 1:1. Atrybuty nowego zbioru encji to: Id wlasciciela, Imie wlasciciela, Nazwisko wlasciciela, Numer telefonu wlasciciela. Kolejnym atrybutem, który musieliśmy rozbić to adres. Występował on w zbiorze encji Pracownicy oraz Parki rozrywki. Atrybuty nowego zbioru encji Adresy to: Miejscowosc, Ulica, Numer budynku, Numer mieszkania (opcjonalny), Kod pocztowy. Ostatnim zbiorem encji który musieliśmy utworzyć to Stanowiska, który wcześniej był atrybutem encji Pracownicy. Nowy zbiór encji zawiera takie atrybuty jak: Id stanowiska, miejsce pracy oraz opis (opcjonalny). Dodaliśmy również kilka nowych dziedzin i reguł (Nadzor rodzica oraz Choroby serca), które w modelu konceptualnym były atrybutami typu boolean. Ostatnią modyfikacją była zmiana związków w zbiorach encji dziedziczących po sobie. Nowe relacje musiały zostać zamienione na relacje 1:1 (obowiązkowy – opcjonalny, gdzie opcjonalny jest zbiór encji, który dziedziczy po rodzicu).

4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym:

Wyjaśnienie:

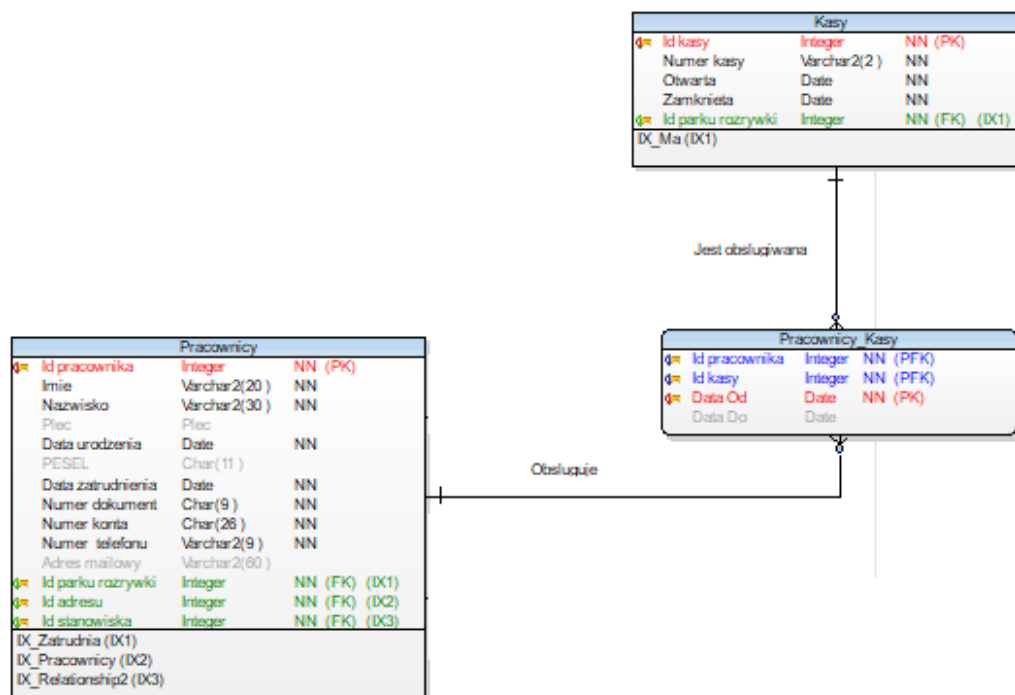
N¹⁾ - Not Null

U²⁾ - Unique

C³⁾ - Check

D⁴⁾ - Default

- Usunięcie związków wiele do wielu (M:N):



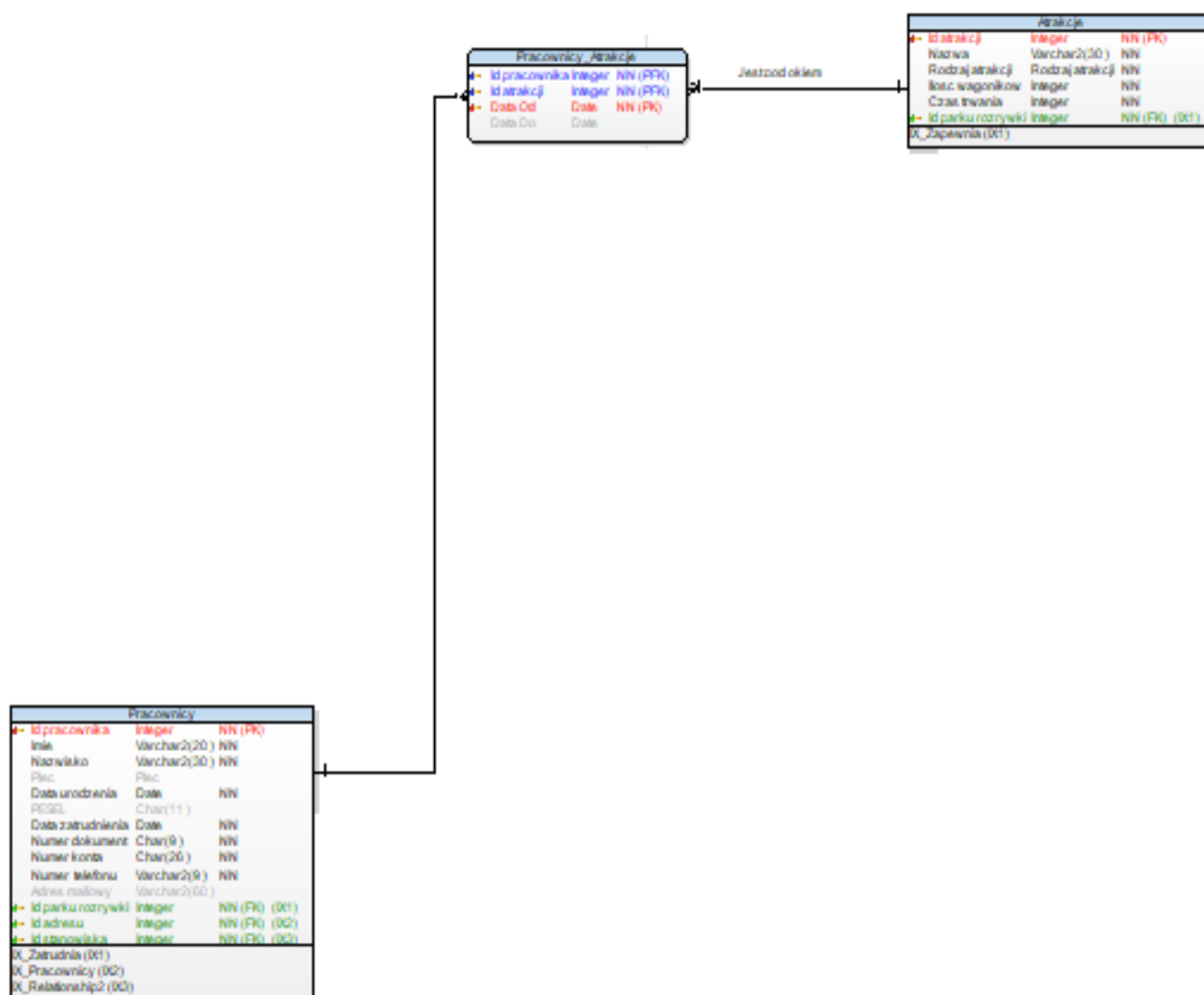
Zbiór encja Pracownicy_Kasy:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PFK	Id_pracownika		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id pracownika
PFK	Id_kasy		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id kasy
PK	Data_Od		Date	YES	NO	NO	NO	Pracownik pracuje na kasie od danej daty
	Data_Do		Date	NO	NO	NO	NO	Pracownik pracuje na kasie do danej daty

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Obsluгуje	Identifying	Pracownicy	Pracownicy_Kasy	1:N
Jest obsluгуwana	Identifying	Kasy	Pracownicy_Kasy	1:N



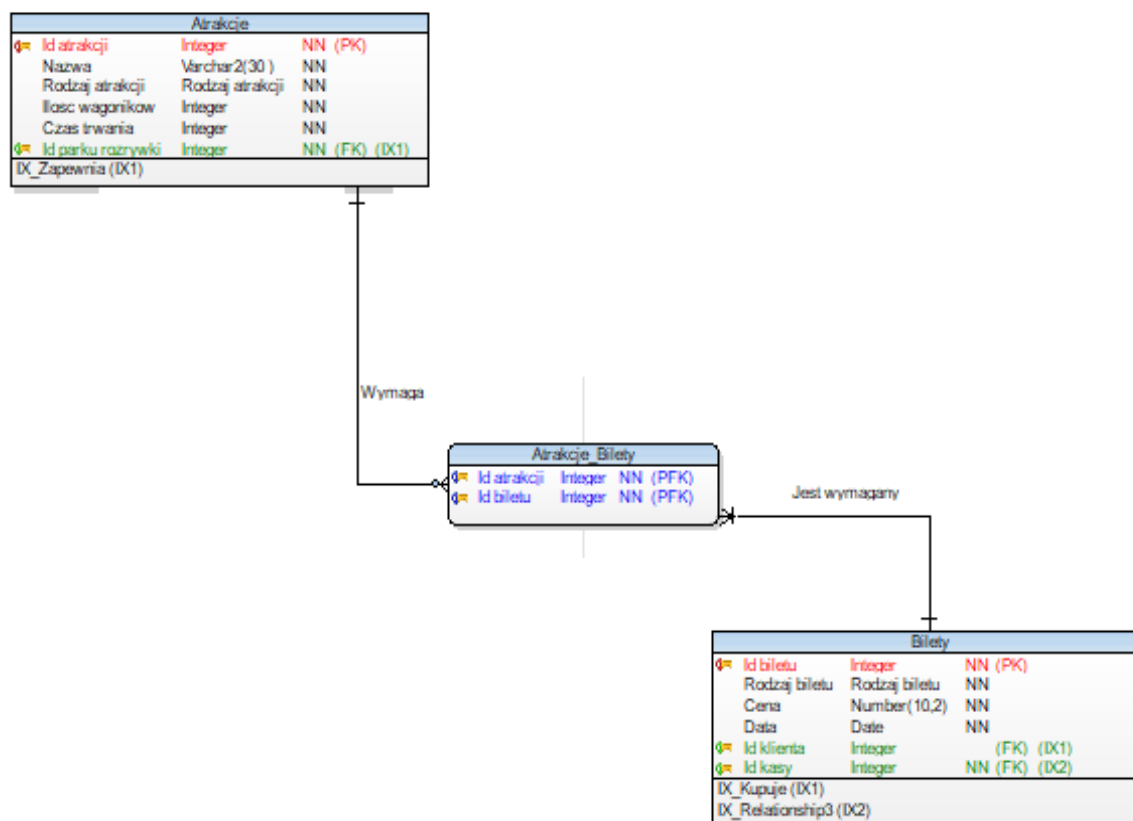
Zbiór encja Pracownicy_Atrakcje:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PFK	Id_pracownika		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id pracownika
PFK	Id_atrakcji		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id atrakcji
PK	Data_Od		Date	YES	NO	NO	NO	Pracownik pracuje przy atrakcji od danej daty
	Data_Do		Date	NO	NO	NO	NO	Pracownik pracuje przy atrakcji do danej daty

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Pracuje przy	Identifying	Pracownicy	Pracownicy Atrakcje	1:N
Jest pod okiem	Identifying	Atrakcje	Pracownicy Atrakcje	1:N



Zbiór encja Atrakcje_Bilety:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PFK	Id_atrakcji		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id atrakcji
PFK	Id_biletu		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id biletu

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Wymaga	Identifying	Atrakcje	Atrakcje_Bilety	1:N
Jest wymagany	Identifying	Bilety	Atrakcje_Bilety	1:N

- Zmiana nazwy encji:

Park rozrywki – Parki rozrywki

Pracownik – Pracownicy

Wynagrodzenie – Wynagrodzenia

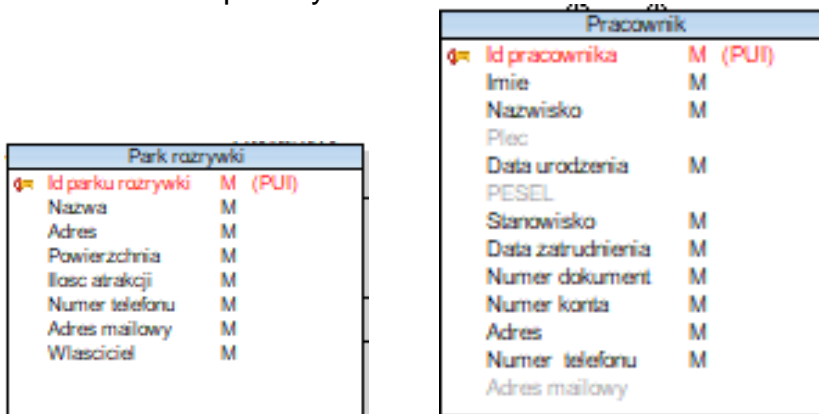
Klient – Klienci

Kasa – Kasy

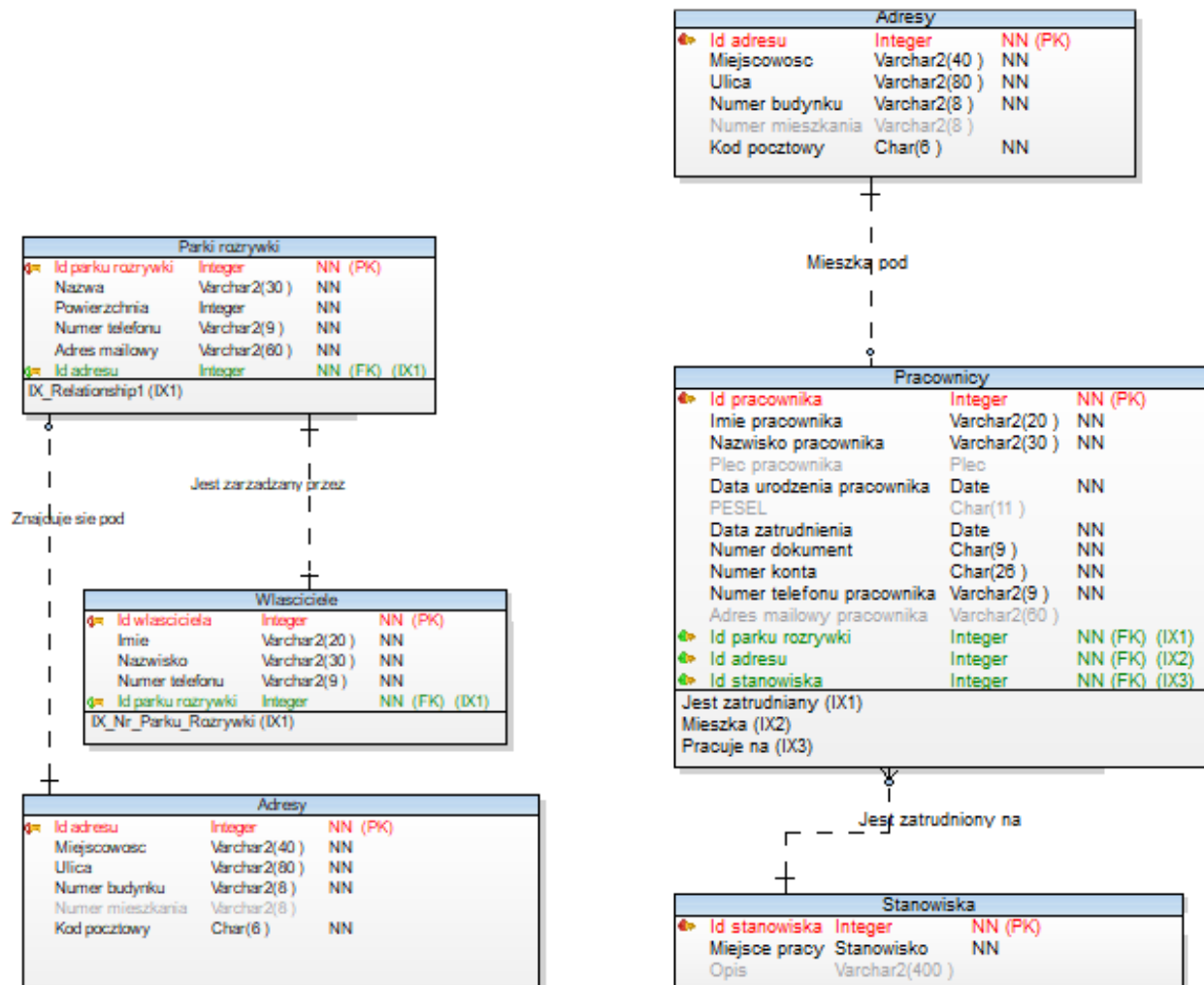
ltd.

- Usunięcie pól segmentowych:

Model konceptualny:



Model logiczny:



Zbiór encji Adresy:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PK	Id_adresu		Integer	YES	NO	NO	NO	Unikatowy numer adresu
	Miejscowosc		Varchar2(40)	YES	NO	NO	NO	Miejscowosc adresu
	Ulica		Varchar2(80)	YES	NO	NO	NO	Ulica adresu
	Numer_budynku		Varchar2(8)	YES	NO	NO	NO	Numer budynku adresu
	Numer_mieszkania		Varchar2(8)	NO	NO	NO	NO	Numer mieszkania adresu
	Kod_pocztowy		Char(6)	YES	NO	NO	NO	Kod pocztowy miejscowosci adresu

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Mieszka_pod	Non-identifying	Adresy	Pracownicy	1:1
Znajduje_sie_pod	Non-identifying	Adresy	Parki_rozrywki	1:1

Zbiór encji Wlasciciele:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PK	Id_wlasciciela		Integer	YES	NO	NO	NO	Unikatowy numer wlasciciela
	Imie_wlasciciela		Varchar2(20)	YES	NO	NO	NO	Imie wlasciciela
	Nazwisko_wlasciciela		Varchar2(30)	YES	NO	NO	NO	Nazwisko wlasciciela
	Numer_telefonu_wlasciciela		Varchar2(9)	YES	NO	NO	NO	Numer telefonu do wlasciciela
FK	Id_parku_rozrywki		Integer	YES	NO	NO	NO	Klucz obcy Id parku rozrywki

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Jest_zarzadzany_przez	Non-identifying	Parki_rozrywki	Wlasciciele	1:1

Zbiór encji Stanowiska:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PK	Id_stanowiska		Integer	YES	NO	NO	NO	Unikatowy numer stanowiska
	Miejsce_pracy	Stanowisko	Varchar2(20)	YES	NO	YES	NO	Miejsce wykonywania danego stanowiska
	Opis		Varchar2(400)	NO	NO	NO	NO	Opis stanowiska

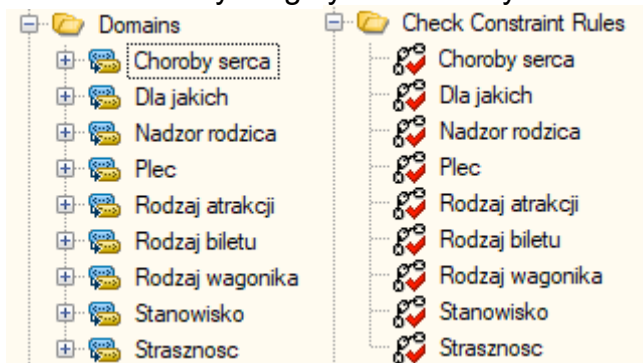
Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Jest_zatrudniony_na	Non-identifying	Stanowiska	Pracownicy	1:N

Domena oraz reguła wyglądają identycznie jak w modelu konceptualnym dla atrybutu stanowisko encji pracownik.

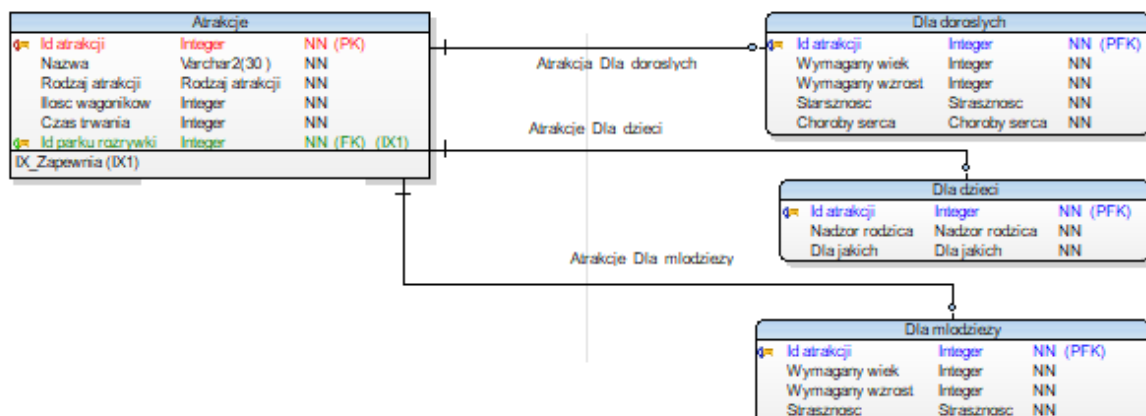
- Nowe dziedziny i reguły:

Nowe dziedziny i reguły to: Choroby serca oraz Nadzor rodzica:



1. Choroby serca - reguła występująca w zbiorze encji dla_doroslych. W modelu konceptualnym był to atrybut typu boolean. Określa czy klient ma problemy z sercem. Odpowiedzi to: 'tak' lub 'nie'.
2. Nadzor rodzica - reguła występująca w zbiorze encji dla_dzieci. W modelu konceptualnym był to atrybut typu boolean. Określa czy do danej atrakcji przeznaczonej dla dzieci wymagany jest nadzór rodzica. Odpowiedzi to: 'tak' lub 'nie'.

- Dziedziczenie:



Zbiór encja Dla_doroslych:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PFK	Id_atrakcji		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id atrakcji
	Wymagany_wiek		Integer	YES	NO	NO	NO	Wymagany wiek aby skorzystac z atrakcji
	Wymagany_wzrost		Integer	YES	NO	NO	NO	Wymagany wzrost aby skorzystac z atrakcji
	Straszność	Straszność	Varchar2(20)	YES	NO	YES	NO	Szraszność atrakcji
	Choroby_serca	Choroby_serca	Varchar2(3)	YES	NO	YES	NO	Czy klient posiada choroby serca

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Atrakcje Dla doroslych	Identifying	Atrakcje	Dla doroslych	1:1

Zbiór encji Dla_dzieci:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PFK	Id_atrakcji		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id atrakcji
	Nadzor_rodzica	Nadzor_rodzica	Varchar2(3)	YES	NO	YES	NO	Czy wymagany jest nadzor rodzica
	Dla_jakich	Dla_jakich	Varchar2(20)	YES	NO	YES	NO	Dla jakich dzieci przeznaczona jest atrakcja

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Atrakcje Dla dzieci	Identifying	Atrakcje	Dla dzieci	1:1

Zbiór encji Dla_mlodziezy:

Attributes

Key	Full Name	Domain	Data Type	N ¹⁾	U ²⁾	C ³⁾	D ⁴⁾	Comments
PFK	Id_atrakcji		Integer	YES	NO	NO	NO	PFK (primary foreign key) Id atrakcji
	Wymagany_wiek		Integer	YES	NO	NO	NO	Wymagany wiek aby skorzystac z atrakcji
	Wymagany_wzrost		Integer	YES	NO	NO	NO	Wymagany wzrost aby skorzystac z atrakcji
	Straszosc	Straszosc	Varchar2(20)	YES	NO	YES	NO	Sztraszosc atrakcji

Relationships

Full Name	Type	Parent Entity	Child Entity	Card.
Atrakcje Dla mlodziezy	Identifying	Atrakcje	Dla mlodziezy	1:1

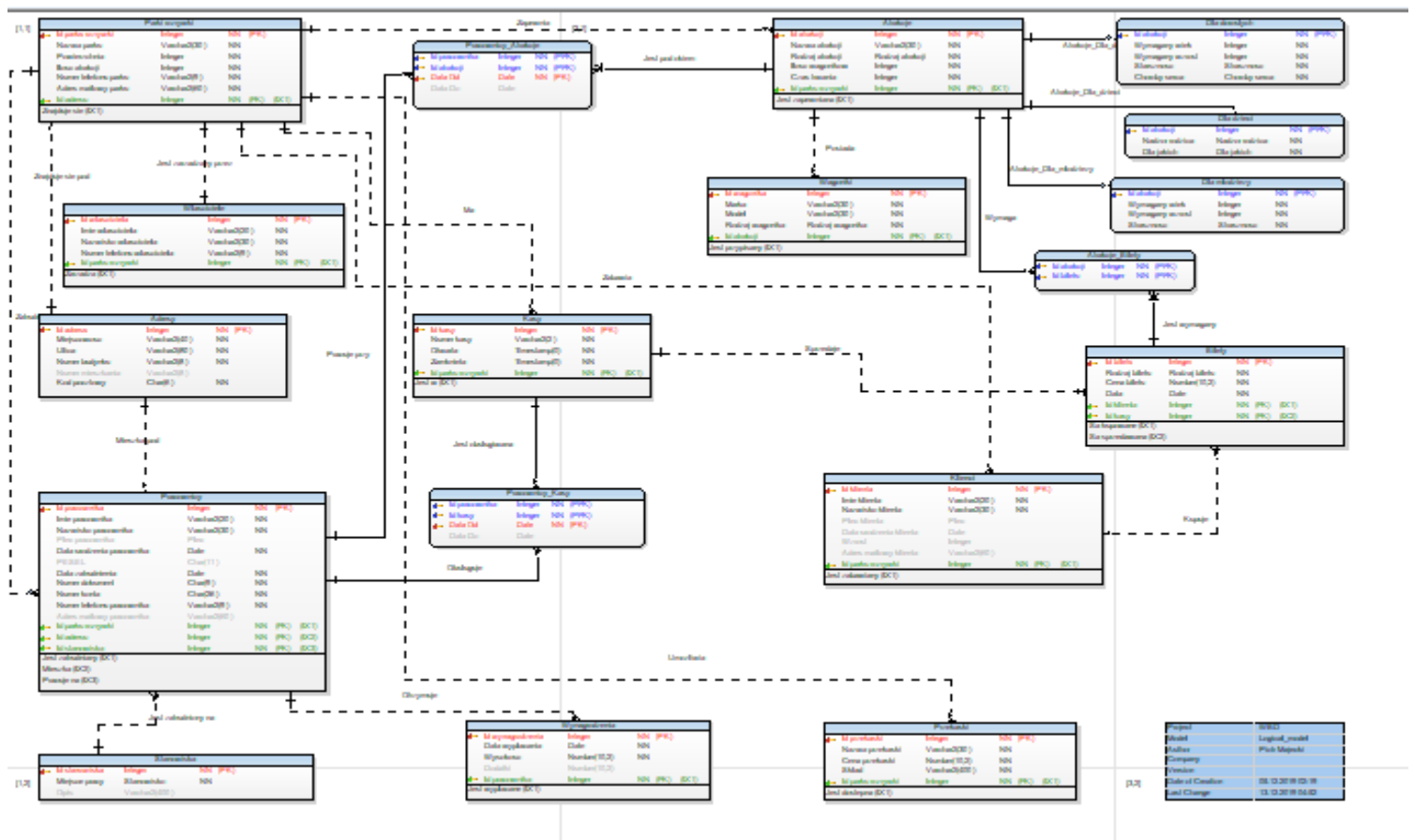
4.3 Proces normalizacji:

Część procesu normalizacji odbyła się już na etapie modelu konceptualnego. Z założeń projektu Nasza baza danych musi znajdować się przynajmniej w 3PN. Zmiany, których musieliśmy dokonać są dokładniej opisane wyżej. Proces normalizacji to:

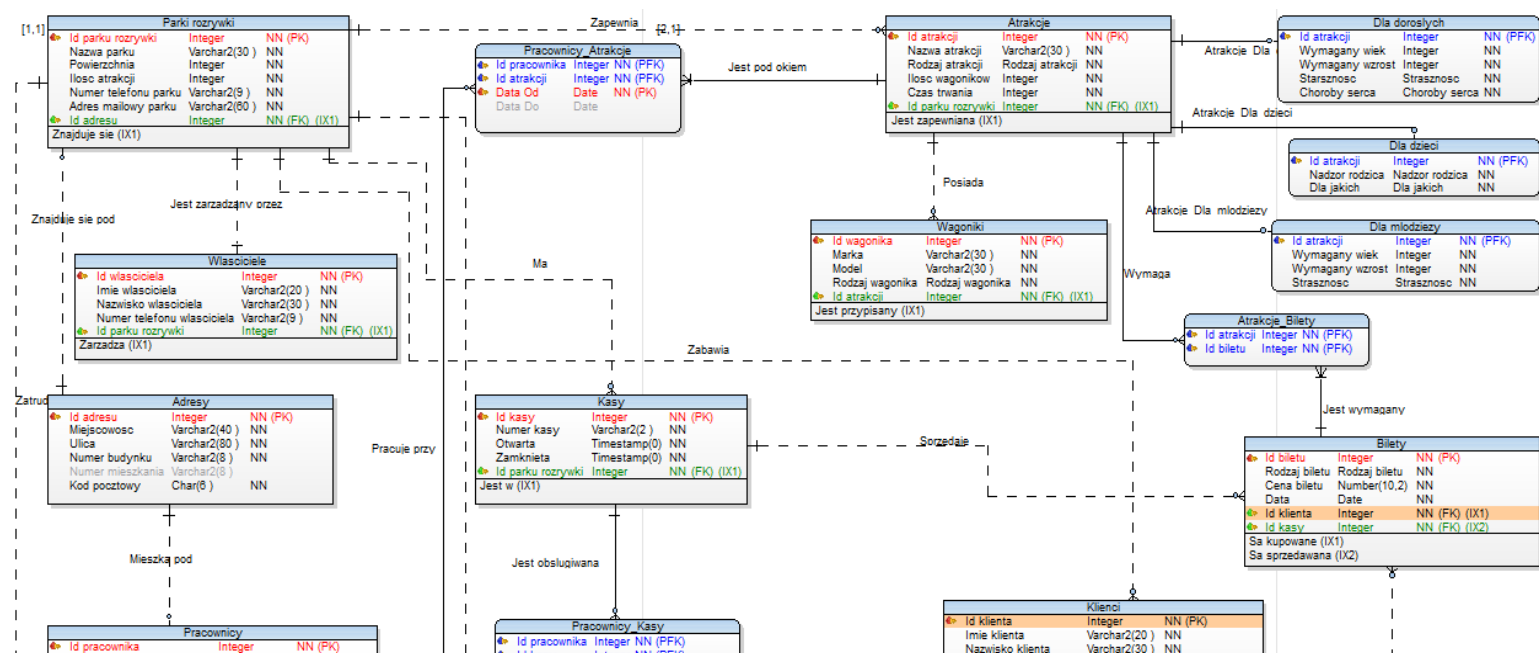
- Każda wartość atrybutu każdego zbioru encji jest wartością elementarną.
- Rozbicie Adresu na wartości atomowe (Miejscowosc, ulica, numer_budynku, numer_mieszkania, kod_pocztowy).
- Utworzenie nowych zbiorów encji: Adresy, Wlasciciele, Stanowiska.
- Wszystkie atrybuty są unikatowe i mają inne nazwy.
- Każdy klucz główny jest prosty.
- Usunięcie związków wiele do wielu (M:N) i zastąpienie ich tablicami bridge'ującymi.

4.4 Schemat ER na poziomie modelu logicznego:

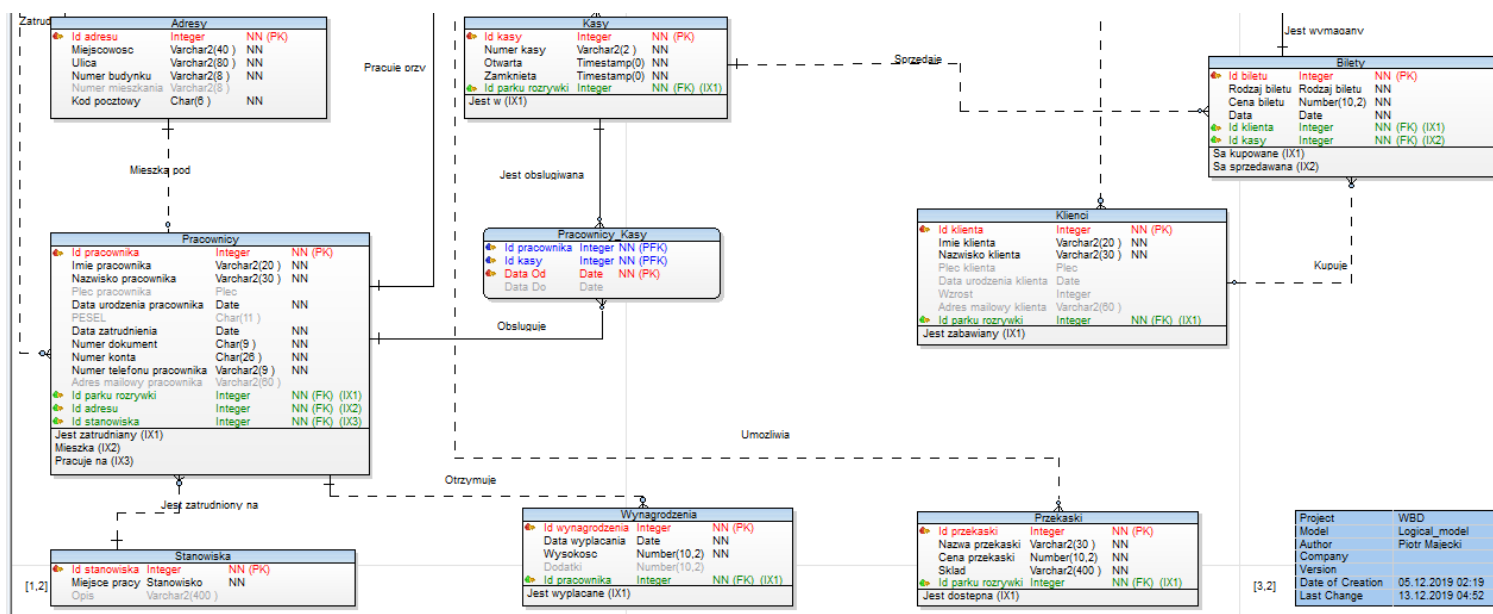
- Całość:



- Góra:



• Dół:



4.5 Więzy integralności:

W celu zabezpieczenia Naszej bazy danych przed błędnym dodaniem danych lub niewłaściwym modyfikowaniem ich, wprowadziliśmy pewne ograniczenia. Większość atrybutów Naszych zbiorów encji jest obowiązkowa (ma wartość NOT NULL). Wszystkie wartości kluczy mają niepowtarzalne wartości (są UNIQUE). Zastosowaliśmy również wiele dziedzin i reguł, aby dokładniej określić zakres danych pól. Dodatkowo, w systemie nie występują pola segmentowe, każdy atrybut ma swoją, niepowtarzalną nazwę, a zbiory encji mają atrybuty o wartościach atomowych, dzięki czemu Nasza baza danych jest bardziej spójna.

4.6 Proces denormalizacji:

Nasza baza danych nie przechowuje wielkiej ilości danych, dlatego zdecydowaliśmy się nie korzystać z procesu denormalizacji. Jak wiadomo tego procesu używamy tylko wtedy, kiedy chcemy zwiększyć wydajność bazy danych, a jest to spowodowane niewystarczającą wydajnością przetwarzania. Przez to tracimy na elastyczności oraz utrudniamy Nasz model (jest bardziej skomplikowany). Według Nas, baza stworzona w projekcie jest wystarczająca wydajna, przez co rezygnujemy z tego procesu.

Jeżeli jednak bylibyśmy zmuszeni do zastosowania denormalizacji to myślę, że zrezygnowalibyśmy z relacji Właściciele i ustawili ją jako atrybuty wielosegmentowy w zbiorze encji Parki rozrywki (tak jak to miało miejsce w modelu konceptualnym).

Moglibyśmy również rozważyć zrezygnowanie ze zbioru encji Adresy, jednak strasznie by się to odbiło na elastyczności struktury, ponieważ nie moglibyśmy dodawać lub modyfikować pojedynczych atrybutów (typu ulica czy numer mieszkania), a musielibyśmy modyfikować, bądź usuwać całe segmenty danych (jakbyśmy przenieśli Nasz park rozrywki na inną ulicę, bądź urząd miasta sam by zmienił nazwę ulicy) to bez denormalizacji, zaktualizowalibyśmy Naszą bazę danych w kilka chwil. Natomiast jeśli adres byłby atrybutem zbioru encji Parki rozrywki to aktualizacja zabrałaby Nam o wiele więcej czasu i nerwów (co może również skutkować większym prawdopodobieństwem, że popełnimy błąd lub literówkę podczas przepisywania danych).

5. Faza fizyczna:

5.1 Projekt transakcji i weryfikacja ich wykonalności:

Transakcje jakie występują w Naszym modelu bazy danych są opisane wyżej (w funkcjonalności systemu – str. 3/4).

Numer	Transakcja/Funkcjonalność	Spełniona?
1	Modyfikacja danych parku rozrywki	TAK
2	Przegląd danych parku rozrywki	TAK
3	Dodawanie/Usuwanie pracowników	TAK
4	Modyfikacja danych pracownika	TAK
5	Przegląd danych pracownika	TAK
6	Określenie wynagrodzenia pracownika	TAK
7	Modyfikacja danych związanych z wynagrodzeniem	TAK
8	Przegląd danych związanych z wynagrodzeniem	TAK
9	Przegląd danych klienta	TAK
10	Modyfikacja danych klienta	TAK
11	Podgląd otwartych kas	TAK
12	Przegląd danych związanych z kasami	TAK
13	Przegląd danych związanych z atrakcjami	TAK
14	Określanie straszności atrakcji	TAK
15	Określanie rodzajów wagoników	TAK
16	Określanie marki i modelu wagoników	TAK
17	Modyfikowanie danych związanych z atrakcjami lub wagonikami	TAK
18	Przegląd danych związanych z przekąskami	TAK
19	Modyfikowanie danych związanych z przekąskami	TAK
20	Przegląd danych związanych z biletami	TAK
21	Modyfikowanie danych związanych z biletami	TAK
22	Określanie rodzajów biletów	TAK
23	Modyfikowanie struktury bazy danych	TAK

Jak widzimy wszystkie założenia z funkcjonalności są spełnione i możemy wykonać każdą z wymienionych transakcji w Naszej bazie danych.

5.2 Strojzenie bazy danych – dobór indeksów:

Stworzone indeksy:

- Parki rozrywki:

Parki rozrywki			Entity Properties						
Id parku rozrywki	Integer	NN (PK)	After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties
Nazwa parku	Varchar2(30)	NN	General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions
Powierzchnia	Integer	NN				Caption	Name	Items	Unique
Ilosc atrakcji	Integer	NN							Status
Numer telefonu parku	Varchar2(9)	NN				Znajduje sie	Znajduje_sie	Id_adresu	<input type="checkbox"/>
Adres mailowy parku	Varchar2(60)	NN							<input checked="" type="checkbox"/>
Id adresu	Integer	NN (FK) (IX1)							
Znajduje sie (IX1)									

- Wlasciciele:

Wlasciciele			Entity Properties						
Id wlasciciela	Integer	NN (PK)	After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties
Imie wlasciciela	Varchar2(20)	NN	General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions
Nazwisko wlasciciela	Varchar2(30)	NN				Caption	Name	Items	Unique
Numer telefonu wlasciciela	Varchar2(9)	NN							Status
Id parku rozrywki	Integer	NN (FK) (IX1)				Zarzadza	Zarzadza	Id_parku_rozrywki	<input type="checkbox"/>
Zarzadza (IX1)									<input checked="" type="checkbox"/>

- Pracownicy:

Pracownicy			Entity Properties						
Id pracownika	Integer	NN (PK)	After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties
Imie pracownika	Varchar2(20)	NN	General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions
Nazwisko pracownika	Varchar2(30)	NN				Caption	Name	Items	Unique
Plec pracownika	Char(1)	NN							Status
Data urodzenia pracownika	Date	NN				Jest zatrudniany	Jest_zatrudniany	Id_parku_rozrywki	<input type="checkbox"/>
PESEL	Char(11)	NN							<input checked="" type="checkbox"/>
Data zatrudnienia	Date	NN				Mieszka	Mieszka	Id_adresu	<input type="checkbox"/>
Numer dokument	Char(9)	NN							<input checked="" type="checkbox"/>
Numer konta	Char(20)	NN				Pracuje na	Pracuje_na	Id_stanowiska	<input type="checkbox"/>
Numer telefonu pracownika	Varchar2(9)	NN							<input checked="" type="checkbox"/>
Adres mailowy pracownika	Varchar2(60)	NN							
Id parku rozrywki	Integer	NN (FK) (IX1)							
Id adresu	Integer	NN (FK) (IX2)							
Id stanowiska	Integer	NN (FK) (IX3)							
Jest zatrudniany (IX1)									
Mieszka (IX2)									
Pracuje na (IX3)									

- Wynagrodzenia:

Wynagrodzenia			Entity Properties						
Id wynagrodzenia	Integer	NN (PK)	After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties
Data wypłacania	Date	NN	General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions
Wysokosc	Number(10,2)	NN				Caption	Name	Items	Unique
Dodatki	Number(10,2)	NN							Status
Id pracownika	Integer	NN (FK) (IX1)				Jest wypłacane	Jest_wypłacane	Id_pracownika	<input type="checkbox"/>
Jest wypłacane (IX1)									<input checked="" type="checkbox"/>

- Przekaski:

Przekaski			Entity Properties						
Id przekaski	Integer	NN (PK)	After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties
Nazwa przekaski	Varchar2(30)	NN	General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions
Cena przekaski	Number(10,2)	NN				Caption	Name	Items	Unique
Sklad	Varchar2(400)	NN							Status
Id parku rozrywki	Integer	NN (FK) (IX1)				Jest dostepna	Jest_dostepna	Id_parku_rozrywki	<input type="checkbox"/>
Jest dostepna (IX1)									<input checked="" type="checkbox"/>

- Kasy:

Kasy			Entity Properties						
Id kasy	Integer	NN (PK)	After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties
Numer kasy	Varchar2(2)	NN	General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions
Otwarta	Timestamp(0)	NN				Caption	Name	Items	Unique
Zamknieta	Timestamp(0)	NN							Status
Id parku rozrywki	Integer	NN (FK) (IX1)				Jest w	Jest_w	Id_parku_rozrywki	<input type="checkbox"/>
Jest w (IX1)									<input checked="" type="checkbox"/>

- Atrakcje:

Atrakcje			After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties						
			General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions	To Do	Before Script				
Id atrakcji	Integer	NN (FK)	Caption		Name		Items		Unique	Status					
Nazwa atrakcji	Varochar2(30)	NN	Jest zapewniana		Jest_zapewniana		Id_parku_rozrywki		<input type="checkbox"/>	<input checked="" type="checkbox"/>					
Rodzaj atrakcji	Rodzaj atrakcji	NN													
Ilosc wagonikow	Integer	NN													
Czas trwania	Integer	NN													
Id parku rozrywki	Integer	NN (FK) (IX1)													
Jest zapewniana (IX1)															

- Wagoniki:

Wagoniki			After Script	Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties								
			General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions	To Do	Before Script						
Id wagonika	Integer	NN (FK)	Caption		Name		Items		Unique	Status							
Marka	Varochar2(30)	NN	Jest przypisany		Jest_przypisany		Id_atrakcji		<input type="checkbox"/>	<input checked="" type="checkbox"/>							
Model	Varochar2(30)	NN															
Rodzaj wagonika	Rodzaj wagonika	NN															
Id atrakcji	Integer	NN (FK) (IX1)															
Jest przypisany (IX1)																	

- Bilety:

Bilety			General	Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions	To Do	Before Script
Id biletu	Integer	NN (PK)	Caption			Name		Items		Unique	Status
Rodzaj biletu	Rodzaj biletu	NN	Sa kupowane			Sa_kupowane		Id_klienta	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Cena biletu	Number(10,2)	NN	Sa sprzedawana			Sa_sprzedawana		Id_kasy	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
Data	Date	NN									
Id klienta	Integer	NN (FK) (IX1)									
Id kasy	Integer	NN (FK) (IX2)									
Sa kupowane (IX1)											
Sa sprzedawana (IX2)											

- Klienci:

Klienci										
Id klienta	Integer	NN (PK)	After Script		Notes	SQL Preview	Relationships	Inner Script	Physical Properties	Table Properties
Imie klienta	Varochar2(20)	NN	General		Attributes	Keys	Indexes	Check Constraints	Triggers	Permissions
Nazwisko klienta	Varochar2(30)	NN							To Do	Before Script
Plec klienta	Plec		Caption		Name			Items	Unique	Status
Data urodzenia klienta	Date		Jest zabawiany		Jest_zabawiany			Id_parku_rozrywki	<input type="checkbox"/>	<input checked="" type="radio"/>
Wzrost	Integer									
Adres mailowy klienta	Varochar2(80)									
Id parku rozrywki	Integer	NN (FK) (IX1)								
Jest zabawiany (IX1)										

5.3 Skrypt SQL zakładający bazę danych:

```
/*
Created: 05.12.2019
Modified: 12.12.2019
Project: WBD
Model: Logical_model
Author: Piotr Majecki
Database: Oracle 19c
*/

-- Create sequences section -----
-----

CREATE SEQUENCE
ParkRozrywkiSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
WlascicielSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE AdresSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
PracownikSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
StanowiskoSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
WynagrodzenieSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE AtrakcjaSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE WagonikSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
DlaDoroslychSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
DlaMlodziezySeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
DlaDzieciSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE BiletSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE KlientSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE KasaSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

CREATE SEQUENCE
PrzekaskiSeq
INCREMENT BY 1
START WITH 1
NOMAXVALUE
NOMINVALUE
CACHE 20;

-- Create tables section -----
-----

-- Table Parki_rozrywki

CREATE TABLE Parki_rozrywki(
  Id_parku_rozrywki Integer NOT
  NULL,
  Nazwa_parku Varchar2(30 ) NOT
  NULL,
  Powierzchnia Integer NOT NULL,
  Ilosc_atrakcji Integer NOT NULL,
  Numer_telefonu_parku
  Varchar2(9 ) NOT NULL,
  Adres_mailowy_parku
  Varchar2(60 ) NOT NULL,
  Id_adresu Integer NOT NULL
);

-- Create indexes for table
Parki_rozrywki

CREATE INDEX Znajduje_sie ON
Parki_rozrywki (Id_adresu);

-- Add keys for table Parki_rozrywki

ALTER TABLE Parki_rozrywki ADD
CONSTRAINT Unique_Identifier1
PRIMARY KEY
(Id_parku_rozrywki);

-- Table and Columns comments
section

COMMENT ON COLUMN
Parki_rozrywki.Id_parku_rozrywki
IS 'Unikatowy numer parku
rozrywki';
COMMENT ON COLUMN
Parki_rozrywki.Nazwa_parku IS
'Nazwa parku rozrywki';
COMMENT ON COLUMN
Parki_rozrywki.Powierzchnia IS
'Powierzchnia parku rozrywki';
COMMENT ON COLUMN
Parki_rozrywki.Ilosc_atrakcji IS
'Ilosc atrakcji w parku rozrywki';
COMMENT ON COLUMN
Parki_rozrywki.Numer_telefonu_par
ku IS 'Numer telefonu do parku
rozrywki';
COMMENT ON COLUMN
Parki_rozrywki.Adres_mailowy_par
ku IS 'Adres e-mail do parku
rozrywki';
COMMENT ON COLUMN
Parki_rozrywki.Id_adresu IS 'Klucz
obcy Id adresu';

-- Table Pracownicy

CREATE TABLE Pracownicy(
  Id_pracownika Integer NOT NULL,
  Imie_pracownika Varchar2(20 )
  NOT NULL,
  Nazwisko_pracownika
  Varchar2(30 ) NOT NULL,
  Plec_pracownika Char(1 ),
  Data_urodzenia_pracownika Date
  NOT NULL,
  PESEL Char(11 ),
  Data_zatrudnienia Date NOT
  NULL,
  Numer_dokument Char(9 ) NOT
  NULL,
  Numer_konta Char(26 ) NOT
  NULL,
  Numer_telefonu_pracownika
  Varchar2(9 ) NOT NULL,
  Adres_mailowy_pracownika
  Varchar2(60 ),
  Id_parku_rozrywki Integer NOT
  NULL,
  Id_adresu Integer NOT NULL,
  Id_stanowiska Integer NOT NULL
);

-- Create indexes for table
Pracownicy
```



```
CREATE INDEX Jest_zatrudniani
ON Pracownicy
(Id_parku_rozrywki);
CREATE INDEX Mieszka ON
Pracownicy (Id_adresu);
CREATE INDEX Pracuje_na ON
Pracownicy (Id_stanowiska);
```

-- Add keys for table Pracownicy

```
ALTER TABLE Pracownicy ADD
CONSTRAINT Unique_Identifier2
PRIMARY KEY (Id_pracownika);
ALTER TABLE Pracownicy ADD
CONSTRAINT Plec_pracownika
CHECK (Plec_pracownika IN
('K','M'));
```

-- Table and Columns comments section

```
COMMENT ON COLUMN
Pracownicy.Id_pracownika IS
'Unikatowy numer pracownika';
COMMENT ON COLUMN
Pracownicy.Imie_pracownika IS
'Imie pracownika';
COMMENT ON COLUMN
Pracownicy.Nazwisko_pracownika
IS 'Nazwisko pracownika';
COMMENT ON COLUMN
Pracownicy.Plec_pracownika IS
'Plec pracownika';
COMMENT ON COLUMN
Pracownicy.Data_urodzenia_praco
wnika IS 'Data urodzenia
pracownika';
COMMENT ON COLUMN
Pracownicy.PESEL IS 'PESEL
pracownika';
COMMENT ON COLUMN
Pracownicy.Data_zatrudnienia IS
'Data zatrudnienia pracownika';
COMMENT ON COLUMN
Pracownicy.Numer_dokument IS
'Numer dokumentu pracownika';
COMMENT ON COLUMN
Pracownicy.Numer_konta IS
'Numer konta bankowego
pracownika';
COMMENT ON COLUMN
Pracownicy.Numer_telefonu_praco
wnika IS 'Numer telefonu
pracownika';
COMMENT ON COLUMN
Pracownicy.Adres_mailowy_praco
wnika IS 'Adres e-mail pracownika';
COMMENT ON COLUMN
Pracownicy.Id_parku_rozrywki IS
'Klucz obcy Id parku rozrywki';
COMMENT ON COLUMN
Pracownicy.Id_adresu IS 'Klucz
obcy Id adresu';
COMMENT ON COLUMN
Pracownicy.Id_stanowiska IS 'Klucz
obcy Id stanowiska';
```

-- Table Wynagrodzenia

```
CREATE TABLE Wynagrodzenia(
    Id_wynagrodzenia Integer NOT
    NULL,
    Data_wypłacania Date NOT
    NULL,
```

```
Wysokosc Number(10,2) NOT
NULL,
Dodatki Number(10,2),
Id_pracownika Integer NOT NULL
);
```

-- Create indexes for table Wynagrodzenia

```
CREATE INDEX Jest_wypłacane
ON Wynagrodzenia
(Id_pracownika);
```

-- Add keys for table Wynagrodzenia

```
ALTER TABLE Wynagrodzenia
ADD CONSTRAINT
Unique_Identifier3 PRIMARY KEY
(Id_wynagrodzenia);
```

-- Table and Columns comments section

```
COMMENT ON COLUMN
Wynagrodzenia.Id_wynagrodzenia
IS 'Unikatowy numer
wynagrodzenia';
COMMENT ON COLUMN
Wynagrodzenia.Data_wypłacania
IS 'Data wypłacania
wynagrodzenia';
COMMENT ON COLUMN
Wynagrodzenia.Wysokosc IS
'Wysokosc wynagrodzenia';
COMMENT ON COLUMN
Wynagrodzenia.Dodatki IS 'Dodatki
do wynagrodzenia';
COMMENT ON COLUMN
Wynagrodzenia.Id_pracownika IS
'Klucz obcy Id pracownika';
```

-- Table Klienci

```
CREATE TABLE Klienci(
    Id_klienta Integer NOT NULL,
    Imie_klienta Varchar2(20 ) NOT
    NULL,
    Nazwisko_klienta Varchar2(30 )
    NOT NULL,
    Plec_klienta Char(1 ),
    Data_urodzenia_klienta Date,
    Wzrost Integer,
    Adres_mailowy_klienta
    Varchar2(60 ),
    Id_parku_rozrywki Integer NOT
    NULL
);
```

-- Create indexes for table Klienci

```
CREATE INDEX Jest_zabawiany
ON Klienci (Id_parku_rozrywki);
```

-- Add keys for table Klienci

```
ALTER TABLE Klienci ADD
CONSTRAINT Unique_Identifier4
PRIMARY KEY (Id_klienta);
ALTER TABLE Klienci ADD
CONSTRAINT Plec_klienta CHECK
(Plec_klienta IN ('K','M'));
```

-- Table and Columns comments section

```
COMMENT ON COLUMN
Klienci.Id_klienta IS 'Unikatowy
numer klienta';
COMMENT ON COLUMN
Klienci.Imie_klienta IS 'Imie klienta';
COMMENT ON COLUMN
Klienci.Nazwisko_klienta IS
'Nazwisko klienta';
COMMENT ON COLUMN
Klienci.Plec_klienta IS 'Plec klienta';
COMMENT ON COLUMN
Klienci.Data_urodzenia_klienta IS
'Data urodzenia klienta';
COMMENT ON COLUMN
Klienci.Wzrost IS 'Wzrost klienta';
COMMENT ON COLUMN
Klienci.Adres_mailowy_klienta IS
'Adres e-mail klienta';
COMMENT ON COLUMN
Klienci.Id_parku_rozrywki IS 'Klucz
obcy Id parku rozrywki';
```

-- Table Atrakcje

```
CREATE TABLE Atrakcje(
    Id_atrakcji Integer NOT NULL,
    Nazwa_atrakcji Varchar2(30 )
    NOT NULL,
    Rodzaj_atrakcji Varchar2(20 )
    NOT NULL,
    Ilosc_wagonikow Integer NOT
    NULL,
    Czas_trwania Integer NOT NULL,
    Id_parku_rozrywki Integer NOT
    NULL
);
```

-- Create indexes for table Atrakcje

```
CREATE INDEX Jest_zapewniana
ON Atrakcje (Id_parku_rozrywki);
```

-- Add keys for table Atrakcje

```
ALTER TABLE Atrakcje ADD
CONSTRAINT Unique_Identifier5
PRIMARY KEY (Id_atrakcji);
ALTER TABLE Atrakcje ADD
CONSTRAINT Rodzaj_atrakcji
CHECK (Rodzaj_atrakcji IN
('dla_dzieci','dla_mlodziwie','dla_do
roslych'));
```

-- Table and Columns comments section

```
COMMENT ON COLUMN
Atrakcje.Id_atrakcji IS 'Unikatowy
numer atrakcji';
COMMENT ON COLUMN
Atrakcje.Nazwa_atrakcji IS 'Nazwa
atrakcji';
COMMENT ON COLUMN
Atrakcje.Rodzaj_atrakcji IS 'Rodzaj
atrakcji';
COMMENT ON COLUMN
Atrakcje.Ilosc_wagonikow IS 'Ilosc
wagonikow w danej atrakcji';
COMMENT ON COLUMN
Atrakcje.Czas_trwania IS 'Czas
trwania atrakcji';
COMMENT ON COLUMN
Atrakcje.Id_parku_rozrywki IS
'Klucz obcy Id parku rozrywki';
```

```

-- Table Bilety

CREATE TABLE Bilety(
    Id_biletu Integer NOT NULL,
    Rodzaj_biletu Varchar2(20 ) NOT
NULL,
    Cena_biletu Number(10,2) NOT
NULL,
    Data Date NOT NULL,
    Id_klienta Integer NOT NULL,
    Id_kasy Integer NOT NULL
);

-- Create indexes for table Bilety

CREATE INDEX Sa_kupowane ON
Bilety (Id_klienta);
CREATE INDEX Sa_sprzedawana
ON Bilety (Id_kasy);

-- Add keys for table Bilety

ALTER TABLE Bilety ADD
CONSTRAINT Unique_Identifier10
PRIMARY KEY (Id_biletu);
ALTER TABLE Bilety ADD
CONSTRAINT Rodzaj_biletu
CHECK (Rodzaj_biletu IN
('jednodniowy','trzydniowy'));

-- Table and Columns comments
section

COMMENT ON COLUMN
Bilety.Id_biletu IS 'Unikatowy numer
biletu';
COMMENT ON COLUMN
Bilety.Rodzaj_biletu IS 'Rodzaj
biletu';
COMMENT ON COLUMN
Bilety.Cena_biletu IS 'Cena biletu';
COMMENT ON COLUMN
Bilety.Data IS 'Data sprzedazy
biletu';
COMMENT ON COLUMN
Bilety.Id_klienta IS 'Klucz obcy Id
klienta';
COMMENT ON COLUMN
Bilety.Id_kasy IS 'Klucz obcy Id
kasy';

-- Table Wagoniki

CREATE TABLE Wagoniki(
    Id_wagonika Integer NOT NULL,
    Marka Varchar2(30 ) NOT NULL,
    Model Varchar2(30 ) NOT NULL,
    Rodzaj_wagonika Varchar2(20 )
NOT NULL,
    Id_atrakcji Integer NOT NULL
);

-- Create indexes for table
Wagoniki

CREATE INDEX Jest_przypisany
ON Wagoniki (Id_atrakcji);

-- Add keys for table Wagoniki

ALTER TABLE Wagoniki ADD
CONSTRAINT Unique_Identifier11
PRIMARY KEY (Id_wagonika);

ALTER TABLE Wagoniki ADD
CONSTRAINT Rodzaj_wagonika
CHECK (Rodzaj_wagonika IN
('szynowy','latajacy','plywajacy','gok
art'));

-- Table and Columns comments
section

COMMENT ON COLUMN
Wagoniki.Id_wagonika IS
'Unikatowy numer wagonika';
COMMENT ON COLUMN
Wagoniki.Marka IS 'Marka
wagonika';
COMMENT ON COLUMN
Wagoniki.Model IS 'Model
wagonika';
COMMENT ON COLUMN
Wagoniki.Rodzaj_wagonika IS
'Rodzaj wagonika w danej atrakcji';
COMMENT ON COLUMN
Wagoniki.Id_atrakcji IS 'Klucz obcy
Id atrakcji';

-- Table Przekaski

CREATE TABLE Przekaski(
    Id_przekaski Integer NOT NULL,
    Nazwa_przekaski Varchar2(30 )
NOT NULL,
    Cena_przekaski Number(10,2)
NOT NULL,
    Sklad Varchar2(400 ) NOT NULL,
    Id_parku_rozrywki Integer NOT
NULL
);

-- Create indexes for table
Przekaski

CREATE INDEX Jest_dostepna
ON Przekaski (Id_parku_rozrywki);

-- Add keys for table Przekaski

ALTER TABLE Przekaski ADD
CONSTRAINT Unique_Identifier12
PRIMARY KEY (Id_przekaski);

-- Table and Columns comments
section

COMMENT ON COLUMN
Przekaski.Id_przekaski IS
'Unikatowy numer przekaski';
COMMENT ON COLUMN
Przekaski.Nazwa_przekaski IS
'Nazwa przekaski';
COMMENT ON COLUMN
Przekaski.Cena_przekaski IS 'Cena
przekaski';
COMMENT ON COLUMN
Przekaski.Sklad IS 'Opis
skladnikow znajdujacych sie w
przekasce';
COMMENT ON COLUMN
Przekaski.Id_parku_rozrywki IS
'Klucz obcy Id parku rozrywki';

-- Table Dla_doroslych

CREATE TABLE Dla_doroslych(
    Id_atrakcji Integer NOT NULL,
    Wymagany_wiek Integer NOT
NULL,
    Wymagany_wzrost Integer NOT
NULL,
    Straszosc Varchar2(20 ) NOT
NULL,
    Choroby_serca Varchar2(3 ) NOT
NULL
);

-- Add keys for table Dla_doroslych

ALTER TABLE Dla_doroslych ADD
CONSTRAINT Unique_Identifier14
PRIMARY KEY (Id_atrakcji);
ALTER TABLE Dla_doroslych ADD
CONSTRAINT Straszosc CHECK
(Straszosc IN
('wcale','troche','straszna','bardzo_s
traszna'));
ALTER TABLE Dla_doroslych ADD
CONSTRAINT Choroby_serca
CHECK (Choroby_serca IN
('tak','nie'));

-- Table and Columns comments
section

COMMENT ON COLUMN
Dla_doroslych.Id_atrakcji IS 'PFK
(primary foreign key) Id atrakcji';
COMMENT ON COLUMN
Dla_doroslych.Wymagany_wiek IS
'Wymagany wiek aby skorzystac z
atrakcji';
COMMENT ON COLUMN
Dla_doroslych.Wymagany_wzrost
IS 'Wymagany wzrost aby
skorzystac z atrakcji';
COMMENT ON COLUMN
Dla_doroslych.Straszosc IS
'Sztraszosc atrakcji';
COMMENT ON COLUMN
Dla_doroslych.Choroby_serca IS
'Czy klient posiada choroby serca';

-- Table Dla_mlodziezy

CREATE TABLE Dla_mlodziezy(
    Id_atrakcji Integer NOT NULL,
    Wymagany_wiek Integer NOT
NULL,
    Wymagany_wzrost Integer NOT
NULL,
    Straszosc Varchar2(20 ) NOT
NULL
);

-- Add keys for table Dla_mlodziezy

ALTER TABLE Dla_mlodziezy ADD
CONSTRAINT Unique_Identifier15
PRIMARY KEY (Id_atrakcji);
ALTER TABLE Dla_mlodziezy ADD
CONSTRAINT
Straszosc_mlodzez CHECK
(Straszosc IN
('wcale','troche','straszna','bardzo_s
traszna'));

-- Table and Columns comments
section

```

```
COMMENT ON COLUMN
Dla_mlodziezy.Id_atrakcji IS 'PFK
(primary foreign key) Id atrakcji';
COMMENT ON COLUMN
Dla_mlodziezy.Wymagany_wiek IS
'Wymagany wiek aby skorzystac z
atrakcji';
COMMENT ON COLUMN
Dla_mlodziezy.Wymagany_wzrost
IS 'Wymagany wzrost aby
skorzystac z atrakcji';
COMMENT ON COLUMN
Dla_mlodziezy.Strasznosc IS
'Sztrasznosc atrakcji';
```

-- Table Dla_dzieci

```
CREATE TABLE Dla_dzieci(
  Id_atrakcji Integer NOT NULL,
  Nadzor_rodzica Varchar2(3 ) NOT
  NULL,
  Dla_jakich Varchar2(20 ) NOT
  NULL
);
```

-- Add keys for table Dla_dzieci

```
ALTER TABLE Dla_dzieci ADD
CONSTRAINT Unique_Identifier16
PRIMARY KEY (Id_atrakcji);
ALTER TABLE Dla_dzieci ADD
CONSTRAINT Nadzor_rodzica
CHECK (Nadzor_rodzica IN
('tak','nie'));
ALTER TABLE Dla_dzieci ADD
CONSTRAINT Dla_jakich CHECK
(Dla_jakich IN
('niemowlak','dziecko','duze_dzieck
o'));
```

-- Table and Columns comments
section

```
COMMENT ON COLUMN
Dla_dzieci.Id_atrakcji IS 'PFK
(primary foreign key) Id atrakcji';
COMMENT ON COLUMN
Dla_dzieci.Nadzor_rodzica IS 'Czy
wymagany jest nadzor rodzica';
COMMENT ON COLUMN
Dla_dzieci.Dla_jakich IS 'Dla jakich
dzieci przeznaczona jest atrakcja';
```

-- Table Kasy

```
CREATE TABLE Kasy(
  Id_kasy Integer NOT NULL,
  Numer_kasy Varchar2(2 ) NOT
  NULL,
  Otwarta Timestamp(0 ) NOT
  NULL,
  Zamkniete Timestamp(0 ) NOT
  NULL,
  Id_parku_rozrywki Integer NOT
  NULL
);
```

-- Create indexes for table Kasy

```
CREATE INDEX Jest_w ON Kasy
(Id_parku_rozrywki);
```

-- Add keys for table Kasy

```
ALTER TABLE Kasy ADD
CONSTRAINT Unique_Identifier17
PRIMARY KEY (Id_kasy);
```

-- Table and Columns comments
section

```
COMMENT ON COLUMN
Kasy.Id_kasy IS 'Unikatowy numer
kasy';
COMMENT ON COLUMN
Kasy.Numer_kasy IS 'Numer kasy';
COMMENT ON COLUMN
Kasy.Otwarta IS 'Czas otwarcia
kasy';
COMMENT ON COLUMN
Kasy.Zamkniete IS 'Czas
zamknietcia kasy';
COMMENT ON COLUMN
Kasy.Id_parku_rozrywki IS 'Klucz
obcy Id parku rozrywki';
```

-- Table Pracownicy_Atrakcje

```
CREATE TABLE
Pracownicy_Atrakcje(
  Id_pracownika Integer NOT NULL,
  Id_atrakcji Integer NOT NULL,
  Data_Od Date NOT NULL,
  Data_Do Date
);
```

-- Table and Columns comments
section

```
COMMENT ON COLUMN
Pracownicy_Atrakcje.Id_pracownik
a IS 'PFK (primary foreign key) Id
pracownika';
COMMENT ON COLUMN
Pracownicy_Atrakcje.Id_atrakcji IS
'PFK (primary foreign key) Id
atrakcji';
COMMENT ON COLUMN
Pracownicy_Atrakcje.Data_Od IS
'Pracownik pracuje przy atrakcji od
danej daty';
COMMENT ON COLUMN
Pracownicy_Atrakcje.Data_Do IS
'Pracownik pracuje przy atrakcji do
danej daty';
```

-- Table Atrakcje_Bilety

```
CREATE TABLE Atrakcje_Bilety(
  Id_atrakcji Integer NOT NULL,
  Id_biletu Integer NOT NULL
);
```

-- Table and Columns comments
section

```
COMMENT ON COLUMN
Atrakcje_Bilety.Id_atrakcji IS 'PFK
(primary foreign key) Id atrakcji';
COMMENT ON COLUMN
Atrakcje_Bilety.Id_biletu IS 'PFK
(primary foreign key) Id biletu';
```

-- Table Wlasciele

```
CREATE TABLE Wlasciele(
  Id_wlasciciela Integer NOT NULL,
```

```
Imie_wlasciciela Varchar2(20 )
NOT NULL,
Nazwisko_wlasciciela
Varchar2(30 ) NOT NULL,
Numer_telefonu_wlasciciela
Varchar2(9 ) NOT NULL,
Id_parku_rozrywki Integer NOT
NULL
);
```

-- Create indexes for table
Wlasciele

```
CREATE INDEX Zarzadza ON
Wlasciele (Id_parku_rozrywki);
```

-- Add keys for table Wlasciele

```
ALTER TABLE Wlasciele ADD
CONSTRAINT PK_Wlasciele
PRIMARY KEY (Id_wlasciciela);
```

-- Table and Columns comments
section

```
COMMENT ON COLUMN
Wlasciele.Id_wlasciciela IS
'Unikatowy numer wlasciciela';
COMMENT ON COLUMN
Wlasciele.Imie_wlasciciela IS
'Imie wlasciciela';
COMMENT ON COLUMN
Wlasciele.Nazwisko_wlasciciela
IS 'Nazwisko wlasciciela';
COMMENT ON COLUMN
Wlasciele.Numer_telefonu_wlasci
ciela IS 'Numer telefonu do
wlasciciela';
COMMENT ON COLUMN
Wlasciele.Id_parku_rozrywki IS
'Klucz obcy Id parku rozrywki';
```

-- Table Adresy

```
CREATE TABLE Adresy(
  Id_adresu Integer NOT NULL,
  Miejscowosc Varchar2(40 ) NOT
  NULL,
  Ulica Varchar2(80 ) NOT NULL,
  Numer_budynku Varchar2(8 )
  NOT NULL,
  Numer_mieszkania Varchar2(8 ),
  Kod_pocztowy Char(6 ) NOT
  NULL
);
```

-- Add keys for table Adresy

```
ALTER TABLE Adresy ADD
CONSTRAINT PK_Adresy
PRIMARY KEY (Id_adresu);
```

-- Table and Columns comments
section

```
COMMENT ON COLUMN
Adresy.Id_adresu IS 'Unikatowy
numer adresu';
COMMENT ON COLUMN
Adresy.Miejscowosc IS
'Miejscowosc adresu';
COMMENT ON COLUMN
Adresy.Ulica IS 'Ulica adresu';
```

```

COMMENT ON COLUMN
Adresy.Numer_budynku IS 'Numer
budynku adresu';
COMMENT ON COLUMN
Adresy.Numer_mieszkania IS
'Numer mieszkania adresu';
COMMENT ON COLUMN
Adresy.Kod_pocztowy IS 'Kod
pocztowy miejscowosci adresu';

-- Table Pracownicy_Kasy

CREATE TABLE
Pracownicy_Kasy(
  Id_pracownika Integer NOT NULL,
  Id_kasy Integer NOT NULL,
  Data_Od Date NOT NULL,
  Data_Do Date
);

-- Add keys for table
Pracownicy_Kasy

ALTER TABLE Pracownicy_Kasy
ADD CONSTRAINT
PK_Pracownicy_Kasy PRIMARY
KEY
(Id_pracownika,Id_kasy,Data_Od);

-- Table and Columns comments
section

COMMENT ON COLUMN
Pracownicy_Kasy.Id_pracownika IS
'PFK (primary foreign key) Id
pracownika';
COMMENT ON COLUMN
Pracownicy_Kasy.Id_kasy IS 'PFK
(primary foreign key) Id kasy';
COMMENT ON COLUMN
Pracownicy_Kasy.Data_Od IS
'Pracownik pracuje na kasie od
danej daty';
COMMENT ON COLUMN
Pracownicy_Kasy.Data_Do IS
'Pracownik pracuje na kasie do
danej daty';

-- Table Stanowiska

CREATE TABLE Stanowiska(
  Id_stanowiska Integer NOT NULL,
  Miejsce_pracy Varchar2(20 ) NOT
  NULL,
  Opis Varchar2(400 )
);

-- Add keys for table Stanowiska

ALTER TABLE Stanowiska ADD
CONSTRAINT PK_Stalowiska
PRIMARY KEY (Id_stanowiska);
ALTER TABLE Stanowiska ADD
CONSTRAINT Miejsce_pracy
CHECK (Miejsce_pracy IN
('kasa','atrakcja'));

-- Table and Columns comments
section

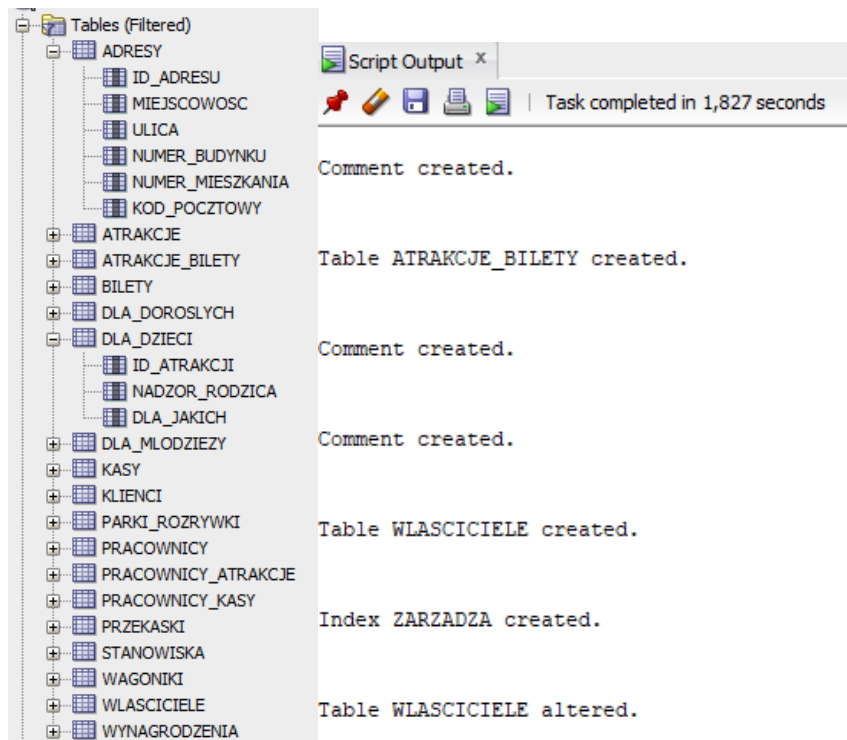
COMMENT ON COLUMN
Stanowiska.Id_stanowiska IS
'Unikatowy numer stanowiska';
COMMENT ON COLUMN
Stanowiska.Miejsce_pracy IS
'Miejsce wykonywania danego
stanowiska';
COMMENT ON COLUMN
Stanowiska.Opis IS 'Opis
stanowiska';

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

ALTER TABLE Pracownicy ADD
CONSTRAINT Zatrudnia FOREIGN
KEY (Id_parku_rozrywki)
REFERENCES Parki_rozrywki
(Id_parku_rozrywki);
ALTER TABLE Wynagrodzenia
ADD CONSTRAINT Otrzymuje
FOREIGN KEY (Id_pracownika)
REFERENCES Pracownicy
(Id_pracownika);
ALTER TABLE Atrakcje ADD
CONSTRAINT Zapewnia FOREIGN
KEY (Id_parku_rozrywki)
REFERENCES Parki_rozrywki
(Id_parku_rozrywki);
ALTER TABLE Klienci ADD
CONSTRAINT Zabawia FOREIGN
KEY (Id_parku_rozrywki)
REFERENCES Parki_rozrywki
(Id_parku_rozrywki);
ALTER TABLE Wagoniki ADD
CONSTRAINT Posiada FOREIGN
KEY (Id_atrakcji) REFERENCES
Atrakcje (Id_atrakcji);
ALTER TABLE Bilety ADD
CONSTRAINT Kupuje FOREIGN
KEY (Id_klienta) REFERENCES
Klienci (Id_klienta);
ALTER TABLE Kasy ADD
CONSTRAINT Ma FOREIGN KEY
(Id_parku_rozrywki) REFERENCES
Parki_rozrywki (Id_parku_rozrywki);
ALTER TABLE Przekaski ADD
CONSTRAINT Umozliwia
FOREIGN KEY (Id_parku_rozrywki)
REFERENCES Parki_rozrywki
(Id_parku_rozrywki);
ALTER TABLE Wlasciciele ADD
CONSTRAINT
Jest_zarzadzany_przez FOREIGN
KEY (Id_parku_rozrywki)
REFERENCES Parki_rozrywki
(Id_parku_rozrywki);
ALTER TABLE Pracownicy ADD
CONSTRAINT Mieszka_pod
FOREIGN KEY (Id_adresu)
REFERENCES Adresy (Id_adresu);
ALTER TABLE Pracownicy_Kasy
ADD CONSTRAINT Obsluguje
FOREIGN KEY (Id_pracownika)
REFERENCES Pracownicy
(Id_pracownika);
ALTER TABLE Pracownicy_Kasy
ADD CONSTRAINT
Jest_obsługiwana FOREIGN KEY
(Id_kasy) REFERENCES Kasy
(Id_kasy);
ALTER TABLE Parki_rozrywki ADD
CONSTRAINT Znajduje_sie_pod
FOREIGN KEY (Id_adresu)
REFERENCES Adresy (Id_adresu);
ALTER TABLE Pracownicy ADD
CONSTRAINT
Jest_zatrudniony_na FOREIGN
KEY (Id_stanowiska)
REFERENCES Stanowiska
(Id_stanowiska);
ALTER TABLE Bilety ADD
CONSTRAINT Sprzedaje
FOREIGN KEY (Id_kasy)
REFERENCES Kasy (Id_kasy);

```

Wszystko się poprawnie wykonało oraz system nie wykrył żadnego błędu.
 Wszystkie sekwencje się dodały.
 Wszystkie tabele się dodały.
 Wszystkie indeksy się dodały.
 Wszystkie klucze główne, klucze obce oraz constrain'y zaktualizowały Nasze tabele.
 Wszystkie komentarze się dodały.



• Skrypt do uzupełnienia wszystkich tabel:

```
INSERT INTO adresy VALUES (1, 'Starachowice','Ostrowiecka', 23, NULL, '27-200');
INSERT INTO adresy VALUES (2, 'Starachowice','Armi Krajowej', 15, 5, '27-200');
INSERT INTO adresy VALUES (3, 'Ostrowiec Swietokrzyski','Kopalniana', 2, NULL, '27-400');
INSERT INTO adresy VALUES (4, 'Skarzysko-Kamienna','Kreta', 12, 12, '27-100');
INSERT INTO adresy VALUES (5, 'Starachowice','Szkolna', 9, 88, '27-200');
INSERT INTO adresy VALUES (6, 'Starachowice','Gorna', 8, 36, '27-200');
INSERT INTO adresy VALUES (7, 'Wachock','Starachowicka', 24, NULL, '27-215');
INSERT INTO adresy VALUES (8, 'Wachock','Mlynarska', 36, NULL, '27-215');
INSERT INTO adresy VALUES (9, 'Starachowice','Opatowa', 5, 15, '27-200');
```

```
INSERT INTO parki_rozrywki VALUES (1, 'Super Park Rozrywki',50000, 82, '123456789', 'super_park_rozrywki@costam.pl', 1);
```

```
INSERT INTO stanowiska VALUES (1, 'kasa', 'Pracuje na kasie nr.1');
INSERT INTO stanowiska VALUES (2, 'atrakcja', 'Obsluguje atrakcje dla mlodziezy Dziki waz');
INSERT INTO stanowiska VALUES (3, 'atrakcja', 'Obsluguje atrakcje dla dzieci Ciuchcia');
INSERT INTO stanowiska VALUES (4, 'atrakcja', 'Obsluguje atrakcje dla dzieci Koniki');
INSERT INTO stanowiska VALUES (5, 'kasa', 'Pracuje na kasie nr.2');
INSERT INTO stanowiska VALUES (6, 'atrakcja', 'Obsluguje atrakcje dla doroslych Wielki mlot');
INSERT INTO stanowiska VALUES (7, 'atrakcja', 'Obsluguje atrakcje dla doroslych Smoczy oddech');
INSERT INTO stanowiska VALUES (8, 'atrakcja', NULL);
```

```
INSERT INTO pracownicy VALUES (1, 'Piotr', 'Majecki', 'M', '1998/01/02', '98010212345', '2018/06/12', 'AZT12A345',
'12345678912345678912345678', '111111111', 'piomaj@costam.pl', 1, 2, 1);
INSERT INTO pracownicy VALUES (2, 'Mateusz', 'Laguna', 'M', '1998/05/09', '98050912345', '2018/06/12', 'XYZ12B345',
'13245678912345678912345678', '222222222', 'mlaguna@costam.pl', 1, 3, 2);
INSERT INTO pracownicy VALUES (3, 'Agata', 'Taka', 'K', '1989/04/12', '89041212345', '2016/04/10', 'ABC12A345',
'12435678912345678912345678', '333333333', 'agataka@costam.pl', 1, 4, 3);
INSERT INTO pracownicy VALUES (4, 'Jan', 'Kowalski', NULL, '1996/08/09', '96080912345', '2017/12/12', 'POI12A345',
'12354768912345678912345678', '444444444', 'kowal@costam.pl', 1, 5, 4);
INSERT INTO pracownicy VALUES (5, 'Martyna', 'Fonta', 'K', '1992/12/01', '92120112345', '2015/03/18', 'TRA12A345',
'12345679812345678912345678', '555555555', 'Martyna.fonta@costam.pl', 1, 6, 5);
INSERT INTO pracownicy VALUES (6, 'Arkadiusz', 'Fila', 'M', '1986/09/22', '86092212345', '2015/06/27', 'TAT12A345',
'12345678921345678912345678', '666666666', 'arekf@costam.pl', 1, 7, 6);
INSERT INTO pracownicy VALUES (7, 'Marlena', 'Faja', 'K', '1995/08/09', '95080912345', '2017/07/18', 'RAK12A345',
'12345678912345678912345678', '777777777', 'fajamarlena@costam.pl', 1, 8, 7);
INSERT INTO pracownicy VALUES (8, 'Filip', 'Kubek', 'M', '1985/05/18', NULL, '2018/09/19', 'NIG12A345',
'12345678912345678912345678', '888888888', NULL, 1, 9, 8);
```

```
INSERT INTO wlasciciele VALUES (1, 'Marek', 'Materac', '987654321', 1);
```

```
INSERT INTO wynagrodzenia VALUES (1, '2019/12/03', 3920, 800, 1);
INSERT INTO wynagrodzenia VALUES (2, '2019/12/03', 3920, 600, 2);
```

```

INSERT INTO wynagrodzenia VALUES (3, '2019/12/03', 5700, NULL, 3);
INSERT INTO wynagrodzenia VALUES (4, '2019/12/03', 5700, 1200, 4);
INSERT INTO wynagrodzenia VALUES (5, '2019/12/03', 6200, NULL, 5);
INSERT INTO wynagrodzenia VALUES (6, '2019/12/03', 6200, NULL, 6);
INSERT INTO wynagrodzenia VALUES (7, '2019/12/03', 5700, 800, 7);
INSERT INTO wynagrodzenia VALUES (8, '2019/12/03', 2890, NULL, 8);

INSERT INTO przekaski VALUES (1, 'Hot-dog', 6, 'Ciepła bułka typ hot-dog z pyszną parówką z indyka i keczupem ', 1);
INSERT INTO przekaski VALUES (2, 'Hamburger', 12, 'Wypieczona duża bułka, z soczystym mięsem wieprzowym, polana sosem 1000 wysp z dodatkiem pomidora, ogorka oraz prażonej cebulki', 1);
INSERT INTO przekaski VALUES (3, 'Zapiekanka', 9, '30cm bułka, z pieczarkami, szynką, serem żółtym, polana keczupem', 1);

INSERT INTO kasy VALUES (1, 1, '2019/12/13 08:00:00', '2019/12/13 14:30:00', 1 );
INSERT INTO kasy VALUES (2, 2, '2019/12/13 14:00:00', '2019/12/13 20:00:00', 1 );

INSERT INTO atrakcje VALUES (1, 'Wielki młot', 'dla_doroslych', 1, 480, 1);
INSERT INTO atrakcje VALUES (2, 'Koniki', 'dla_dzieci', 5, 300, 1);
INSERT INTO atrakcje VALUES (3, 'Dziki waz', 'dla_mlodziuzy', 10, 300, 1);
INSERT INTO atrakcje VALUES (4, 'Ciuchcia', 'dla_dzieci', 8, 600, 1);
INSERT INTO atrakcje VALUES (5, 'Smoczy oddech', 'dla_doroslych', 10, 360, 1);

INSERT INTO wagoniki VALUES (1, 'Mercedes', 'F2019', 'latajacy', 1);
INSERT INTO wagoniki VALUES (2, 'Floki', 'DOK98', 'gokart', 2);
INSERT INTO wagoniki VALUES (3, 'Dandan', 'T5', 'plywajacy', 3);
INSERT INTO wagoniki VALUES (4, 'Ferari', 'Nitro26', 'szynowy', 4);
INSERT INTO wagoniki VALUES (5, 'Polonus', 'adik2', 'szynowy', 5);

INSERT INTO klienci VALUES (1, 'Franciszek', 'Zawada', 'M', '1972/08/09', 182, 'fransiszawada@costam.pl', 1);
INSERT INTO klienci VALUES (2, 'Anna', 'Maciag', 'K', NULL, 175, 'a.maciag@costam.pl', 1);
INSERT INTO klienci VALUES (3, 'Sylwester', 'Migus', 'M', '1968/10/10', NULL, 'sylwester.migus@costam.pl', 1);
INSERT INTO klienci VALUES (4, 'Krystyna', 'Pajak', NULL, '2002/06/24', 175, 'krysia.pajak@costam.pl', 1);
INSERT INTO klienci VALUES (5, 'Adam', 'Wiercpieta', 'M', '2012/10/02', 145, NULL, 1);

INSERT INTO bilety VALUES (1, 'jednodniowy', 20, '2019/12/10', 1, 1);
INSERT INTO bilety VALUES (2, 'trzydniowy', 40, '2019/12/12', 2, 1);
INSERT INTO bilety VALUES (3, 'jednodniowy', 20, '2019/12/10', 3, 2);
INSERT INTO bilety VALUES (4, 'jednodniowy', 12, '2019/12/10', 4, 1);
INSERT INTO bilety VALUES (5, 'jednodniowy', 8, '2019/12/10', 5, 2);

INSERT INTO atrakcje_bilety VALUES (1, 2);
INSERT INTO atrakcje_bilety VALUES (2, 5);
INSERT INTO atrakcje_bilety VALUES (3, 4);
INSERT INTO atrakcje_bilety VALUES (4, 5);
INSERT INTO atrakcje_bilety VALUES (5, 1);
INSERT INTO atrakcje_bilety VALUES (1, 3);

INSERT INTO dla_doroslych VALUES (1, 18, 170, 'straszna', 'nie');
INSERT INTO dla_doroslych VALUES (5, 18, 180, 'bardzo_straszna', 'nie');

INSERT INTO dla_dzieci VALUES (2, 'tak', 'dziecko');
INSERT INTO dla_dzieci VALUES (4, 'nie', 'duze_dziecko');

INSERT INTO dla_mlodziuzy VALUES (3, 12, 160, 'troche');

INSERT INTO pracownicy_atrakcje VALUES (2, 1, '2018/06/12', NULL);
INSERT INTO pracownicy_atrakcje VALUES (3, 2, '2017/01/01', '2020/01/01');
INSERT INTO pracownicy_atrakcje VALUES (4, 3, '2017/12/12', NULL);
INSERT INTO pracownicy_atrakcje VALUES (6, 4, '2016/01/01', '2020/01/01');
INSERT INTO pracownicy_atrakcje VALUES (7, 5, '2017/07/18', NULL);

INSERT INTO pracownicy_kasy VALUES (1, 1, '2018/06/12', NULL);
INSERT INTO pracownicy_kasy VALUES (5, 2, '2016/01/01', NULL);

```

Wszystko się poprawnie wykonało oraz system nie wykrył żadnego błędu.
Wszystkie tabele się zaktualizowały i dane zostały dodane.

Script Output x
Task completed in 1,942 seconds
1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

1 row inserted.

ID_PRACOWNIKA	ID_ATRAKCJI	DATA_OD	DATA_DO
1	2	18/06/12	(null)
2	3	17/01/01	20/01/01
3	4	17/12/12	(null)
4	6	16/01/01	20/01/01
5	7	17/07/18	(null)

Messages - Log

Messages Logging Page Statements x

Oracle Connections

SYSTEM

WBD_PROJEKT

WND_PROJEKT_CZ1

Tables (Filtered)

ADRESY

ATRAKCJE

ATRAKCJE_BILETY

BILETY

DLA_DOROSLYCH

DLA_DZIECI

DLA_MLODZIEZY

KASY

KLIENCI

ID_ADRESU	MIJSCOWOSC	ULICA	NUMER_BUDYNKU	NUMER_MIESZKANIA	KOD_POCZTOWY
1	1 Starachowice	Ostrowiecka	23	(null)	27-200
2	2 Starachowice	Armi Krajowej	15	5	27-200
3	3 Ostrowiec Swietokrzyski	Kopalniana	2	(null)	27-400
4	4 Skarzysko-Kamienna	Kreta	12	12	27-100
5	5 Starachowice	Szkolna	9	88	27-200
6	6 Starachowice	Gorna	8	36	27-200
7	7 Wachock	Starachowicka	24	(null)	27-215
8	8 Wachock	Mlynarska	36	(null)	27-215
9	9 Starachowice	Opatowa	5	15	27-200

5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych:

- Wypisz wszystko o pracownikach:

SELECT * FROM pracownicy;

98 SELECT * FROM pracownicy;

Script Output x Query Result x

SQL All Rows Fetched: 8 in 0,007 seconds

ID_PRACOWNIKA	IMIE_PRACOWNIKA	NAZWISKO_PRACOWNIKA	PLEC_PRACOWNIKA	DATA_URODZENIA_PRACOWNIKA	PESEL	DATA_ZATRUDNIENIA	NUMER_DOKUMENT	NUMER
1	Piotr	Majecki	M	98/01/02	98010212345	18/06/12	AZT12A345	1234567
2	Mateusz	Laguna	M	98/05/09	98050912345	18/06/12	XYZ12B345	1324567
3	Agata	Taka	K	89/04/12	89041212345	16/04/10	ABC12A345	1243567
4	Jan	Kowalski	(null)	96/08/09	96080912345	17/12/12	POI12A345	1235476
5	Martyna	Fonta	K	92/12/01	92120112345	15/03/18	TRA12A345	1234567
6	Arkadiusz	Fila	M	86/09/22	86092212345	15/06/27	TAT12A345	1234567
7	Marlena	Faja	K	95/08/09	95080912345	17/07/18	RAK12A345	1234567
8	Filip	Kubek	M	85/05/18	(null)	18/09/19	NIG12A345	1234567

- Wypisz wszystkie informacje o pracownikach, którzy pracują na stanowisku atrakcja:

```
SELECT * FROM pracownicy
NATURAL JOIN stanowiska where miejsce_pracy = 'atrakcja';
```

100 SELECT * FROM pracownicy
101 NATURAL JOIN stanowiska where miejsce_pracy = 'atrakcja';
102

Script Output x Query Result x

All Rows Fetched: 6 in 0,017 seconds

_STANOWISKA	ID_PRACOWNIKA	IMIE_PRACOWNIKA	NAZWISKO_PRACOWNIKA	PLEC_PRACOWNIKA	DATA_URODZENIA_PRACOWNIKA	PESEL	DATA_ZATRUDNIENIA	NUMER_DOKU
1	2	2 Mateusz	Laguna	M	98/05/09	98050912345	18/06/12	XYZ12B345
2	3	3 Agata	Taka	K	89/04/12	89041212345	16/04/10	ABC12A345
3	4	4 Jan	Kowalski	(null)	96/08/09	96080912345	17/12/12	POI12A345
4	6	6 Arkadiusz	Fila	M	86/09/22	86092212345	15/06/27	TAT12A345
5	7	7 Marlena	Faja	K	95/08/09	95080912345	17/07/18	RAK12A345
6	8	8 Filip	Kubek	M	85/05/18	(null)	18/09/19	NIG12A345

- Wypisz wysokość wynagrodzenia pracownika, którego imię to Piotr, a nazwisko to Majecki:

```
SELECT wysokosc FROM wynagrodzenia
NATURAL JOIN pracownicy WHERE
(pracownicy.imie_pracownika = 'Piotr' AND pracownicy.nazwisko_pracownika = 'Majecki');
```

.03	SELECT wysokosc FROM wynagrodzenia
.04	NATURAL JOIN pracownicy WHERE
.05	(pracownicy.imie_pracownika = 'Piotr' AND pracownicy.nazwisko_pracownika = 'Majecki');
.06	

Script Output	Query Result
All Rows Fetched: 1 in 0,022 seconds	
WYSOKOSC	
1	3920

- Wypisz wszystkie informacje o atrakcjach, które korzystają z wagonika, którego rodzaj to latający lub szynowy:

```
SELECT * FROM atrakcje
NATURAL JOIN wagoniki where rodzaj_wagonika IN ('latajacy','szynowy');
```

07 SELECT * FROM atrakcje
08 NATURAL JOIN wagoniki where rodzaj_wagonika IN ('latajacy','szynowy');
09

Script Output x Query Result x

All Rows Fetched: 3 in 0,021 seconds

ID_ATRAKCJI	NAZWA_ATRAKCJI	RODZAJ_ATRAKCJI	ILOSC_WAGONIKOW	CZAS_TRWANIA	ID_PARKU_ROZRYWKI	ID_WAGONIKA	MARKA	MODEL	RODZAJ_WAGONIKA
1	1 Wielki mlot	dla_doroslych	1	480	1	1	Mercedes	F2019	latajacy
2	4 Ciuchcia	dla_dzieci	8	600	1	4	Ferari	Nitro26	szynowy
3	5 Smoczy oddech	dla_doroslych	10	360	1	5	Polonus	adik2	szynowy

- Wypisz takie dane jak imię klienta, nazwisko klienta, data urodzenia klienta dla klienta o Id_klienta równym 4:

```
SELECT imie_klienta, nazwisko_klienta, data_urodzenia_klienta FROM klienci
WHERE klienci.id_klienta = 4;
```



```
.09 SELECT imie_klienta, nazwisko_klienta, data_urodzenia_klienta FROM klienci
.10
.11 WHERE klienci.id_klienta = 4;
.12
```

Script Output x Query Result x

All Rows Fetched: 1 in 0,011 seconds

IMIE_KLIENTA	NAZWISKO_KLIENTA	DATA_URODZENIA_KLIENTA
1 Krystyna	Pajak	02/06/24

- Wypisz wszystkie informacje o atrakcjach dla dorosłych, w których wymagany wzrost jest pomiędzy 165cm, a 175cm:

SELECT * FROM Dla_doroslych WHERE wymagany_wzrost BETWEEN 165 AND 175;

```
112
113 SELECT * FROM Dla_doroslych WHERE wymagany_wzrost BETWEEN 165 AND 175;
114
```

Script Output x Query Result x

All Rows Fetched: 1 in 0,011 seconds

ID_ATRAKCJI	WYMAGANY_WIEK	WYMAGANY_WZROST	STRASZNOSC	CHOROBY_SERCA
1	1	18	170 straszna	nie

6. Bibliografia

1. Slajdy wykładowe do przedmiotu Wprowadzenie do Baz Danych prowadzonego przez dr inż. Marcina Kowalczyka w semestrze 19Z.
2. Dokumentacje użytych programów.
3. YouTube - kanał „nieinformatyk”, kurs SQL’a.