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](#_Toc27418262)

[2.Definicja systemu: 3](#_Toc27418263)

[2.1 Perspektywy użytkowników: 4](#_Toc27418265)

[3. Model konceptualny: 6](#_Toc27418266)

[3.1 Definicja zbiorów encji określonych w projekcie: 6](#_Toc27418267)

[3.2 Ustalenie związków między encjami i ich typów: 7](#_Toc27418268)

[3.3 Określenie atrybutów i ich dziedzin: 9](#_Toc27418269)

[3.4 Dodatkowe reguły integralnościowe: 14](#_Toc27418270)

[3.5 Klucze kandydujące i główne: 15](#_Toc27418271)

[3.6 Schemat ER na poziomie konceptualnym: 16](#_Toc27418272)

[3.7 Problem pułapek szczelinowych i wachlarzowych: 16](#_Toc27418273)

[4. Model logiczny: 18](#_Toc27418274)

[4.1 Charakterystyka modelu relacyjnego: 18](#_Toc27418275)

[4.2 Usunięcie właściwości niekompatybilnych z modelem relacyjnym: 19](#_Toc27418276)

[4.3 Proces normalizacji: 25](#_Toc27418277)

[4.4 Schemat ER na poziomie modelu logicznego: 26](#_Toc27418278)

[4.5 Więzy integralności: 27](#_Toc27418279)

[4.6 Proces denormalizacji: 27](#_Toc27418280)

[5. Faza fizyczna: 29](#_Toc27418281)

[5.1 Projekt transakcji i weryfikacja ich wykonalności: 29](#_Toc27418282)

[5.2 Strojenie bazy danych – dobór indeksów: 30](#_Toc27418283)

[5.3 Skrypt SQL zakładający bazę danych: 32](#_Toc27418284)

[5.4 Przykłady zapytań i poleceń SQL odnoszących się do bazy danych: 39](#_Toc27418286)

[6. Bibliografia 41](#_Toc27418287)

# **Zakres i cel projektu:**

* Cel projektu:

Celem projektu było zapoznanie się z projektowaniem i tworzeniem baz danych. Należało zaprojektować relacyjną bazę danych na poziomie konceptualnym, 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.

# **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 orasz 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 konceptualnego lub logicznego wykonanego w TOAD Data Modelerze.

Wszystkie związki pomiędzy encjami są binarne.

* Związki encji park rozrywki:

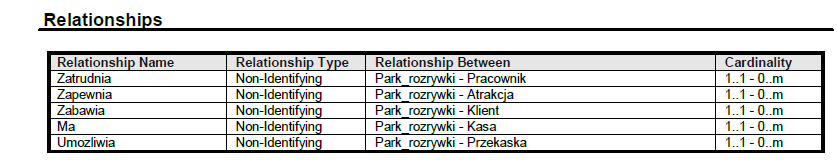
Park rozrywki – Pracownik: obowiązkowy – opcjonalny

Park rozrywki – Atrakcja: obowiązkowy – opcjonalny

Park rozrywki – Klient: obowiązkowy – opcjonalny

Park rozrywki – Kasa: obowiązkowy – opcjonalny

Park rozrywki – Przekaska: obowiązkowy – opcjonalny



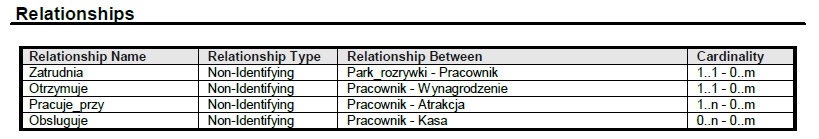
* Związki encji pracownik:

Pracownik – Park rozrywki: opcjonalny – obowiązkowy

Pracownik – Wynagrodzenie: obowiązkowy – opcjonalny

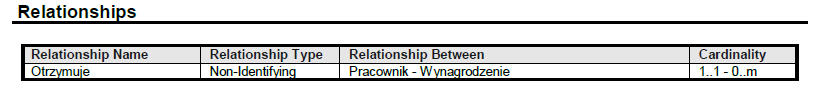
Pracownik – Atrakcja: obowiązkowy – opcjonalny

Pracownik – Kasa: opcjonalny – opcjonalny



* Związki encji wynagrodzenie:

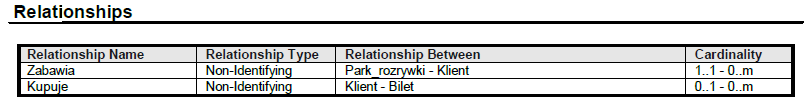
Wynagrodzenie – Pracownik: opcjonalny – obowiązkowy



* Związki encji klient:

Klient – Park rozrywki: opcjonalny – obowiązkowy

Klient – Bilet: opcjonalny – opcjonalny



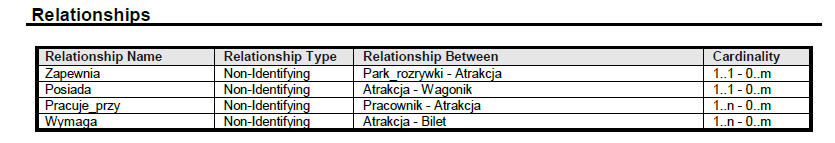
* Związki encji atrakcja:

Atrakcja – Park rozrywki: opcjonalny – obowiązkowy

Atrakcja – Wagonik: obowiązkowy – opcjonalny

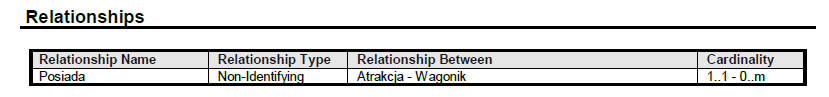
Atrakcja – Pracownik: opcjonalny – obowiązkowy

Atrakcja – Bilet: obowiązkowy – opcjonalny



* Związki encji wagonik:

Wagonik – Atrakcja: opcjonalny – obowiązkowy

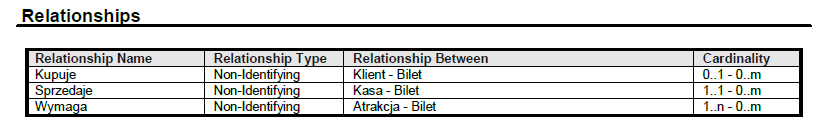


* Związki encji bilet:

Bilet – Klient: opcjonalny – opcjonalny

Bilet – Kasa: opcjonalny – obowiązkowy

Bilet – Atrakcja: opcjonalny – obowiązkowy

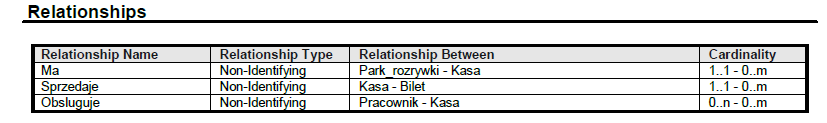


* Związki encji kasa:

Kasa – Park rozrywki: opcjonalny – obowiązkowy

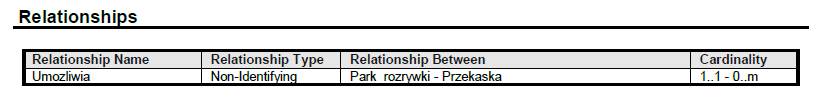
Kasa – Bilet: obowiązkowy – opcjonalny

Kasa – Pracownik: opcjonalny – opcjonalny



* Związki encji przekąski:

Przekaska – park rozrywki: opcjonalny – obowiązkowy



# 3.3 Określenie atrybutów i ich dziedzin:

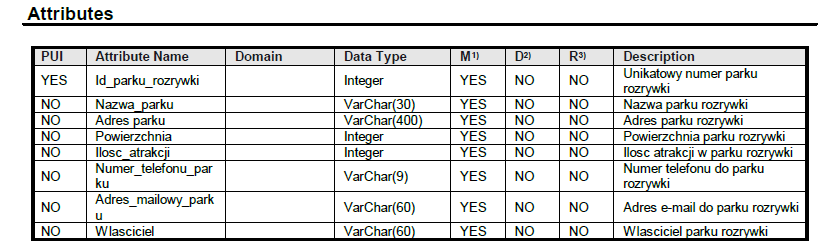
Wyjaśnienie:

M1) - Mandatory

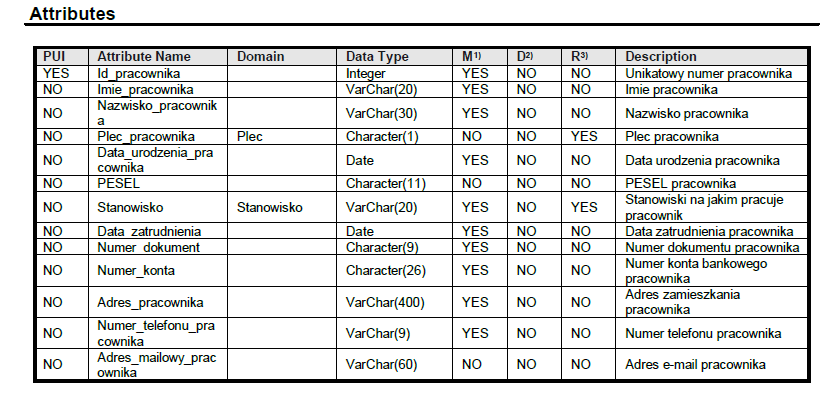
D2) - Default

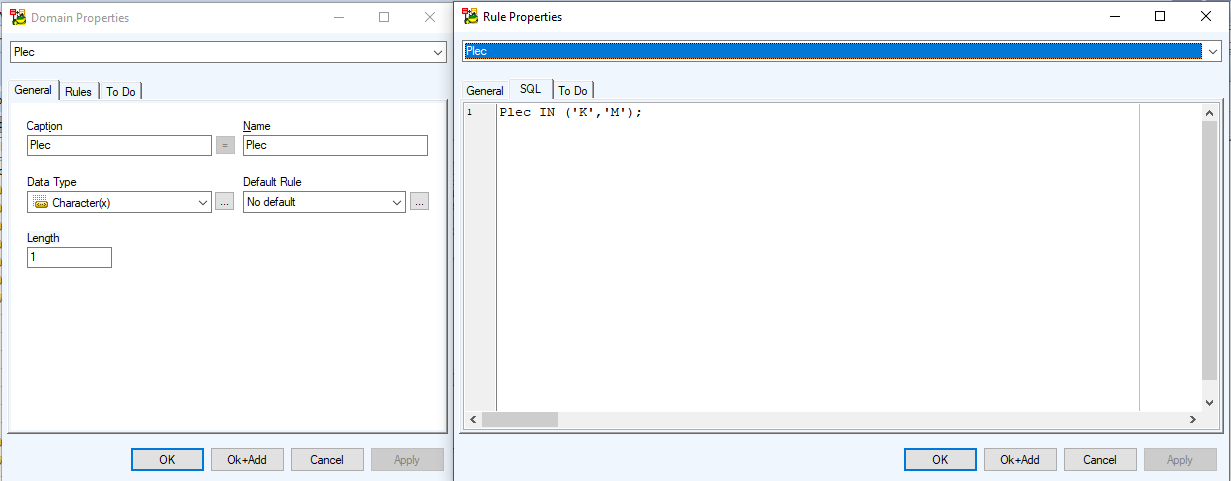
R3) - Rule

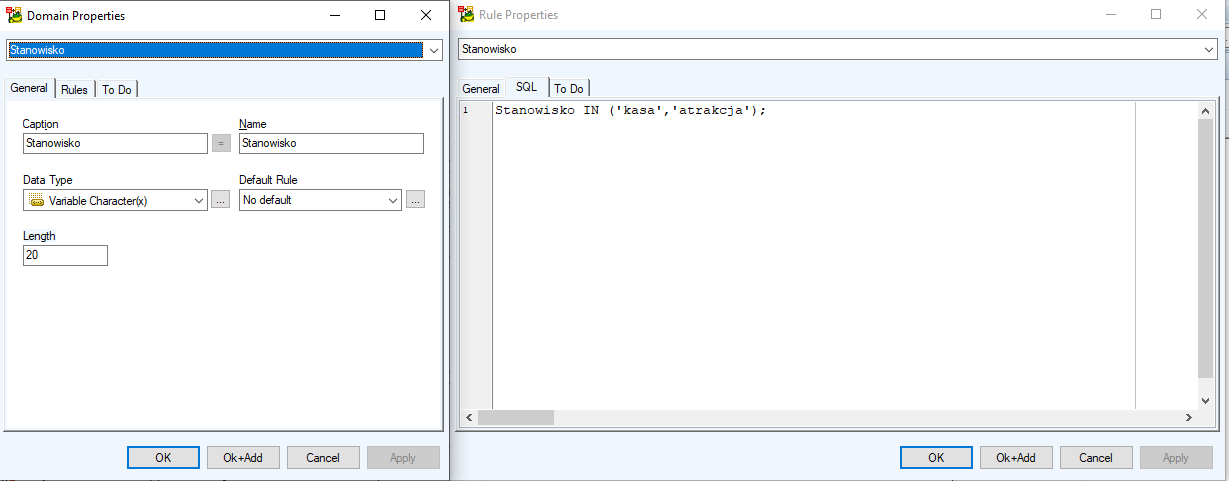
* Encja park rozrywki:



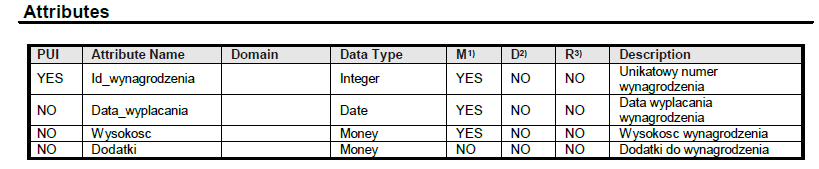
* Encja pracownik:



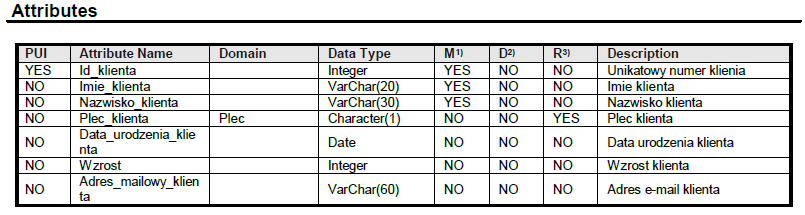




* Encja wynagrodzenie:

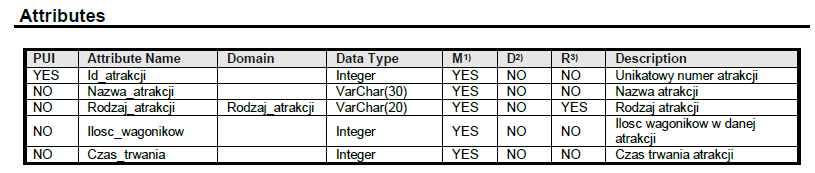


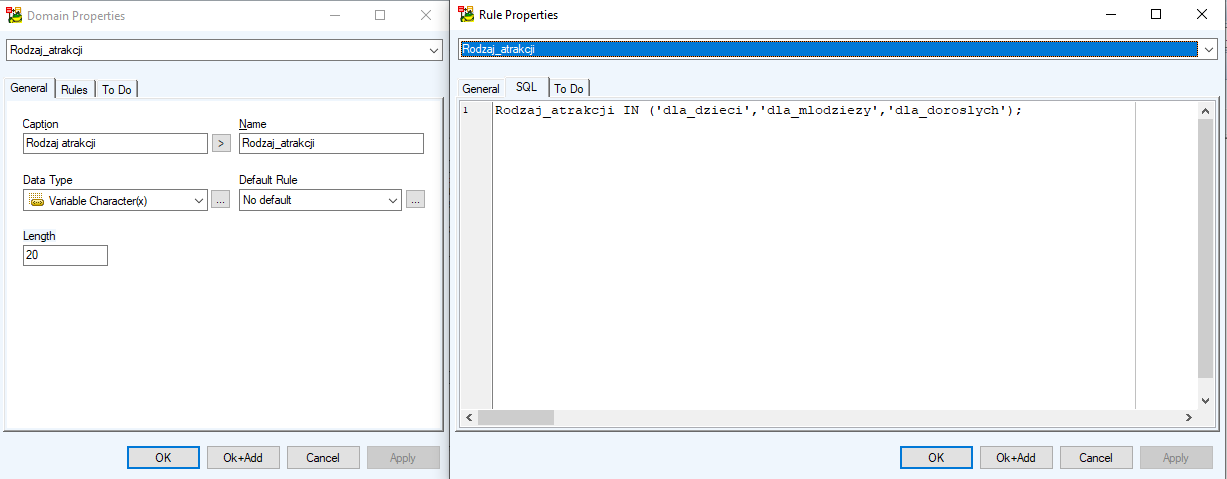
* Encja klient:



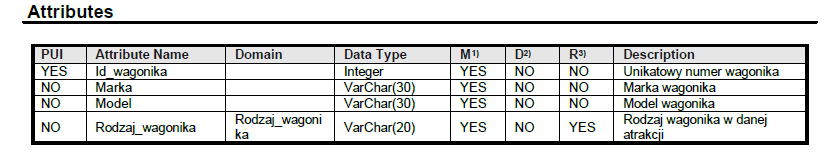
Domena i reguła odnosząca się do Plec znajduje się wyżej (encja pracownik).

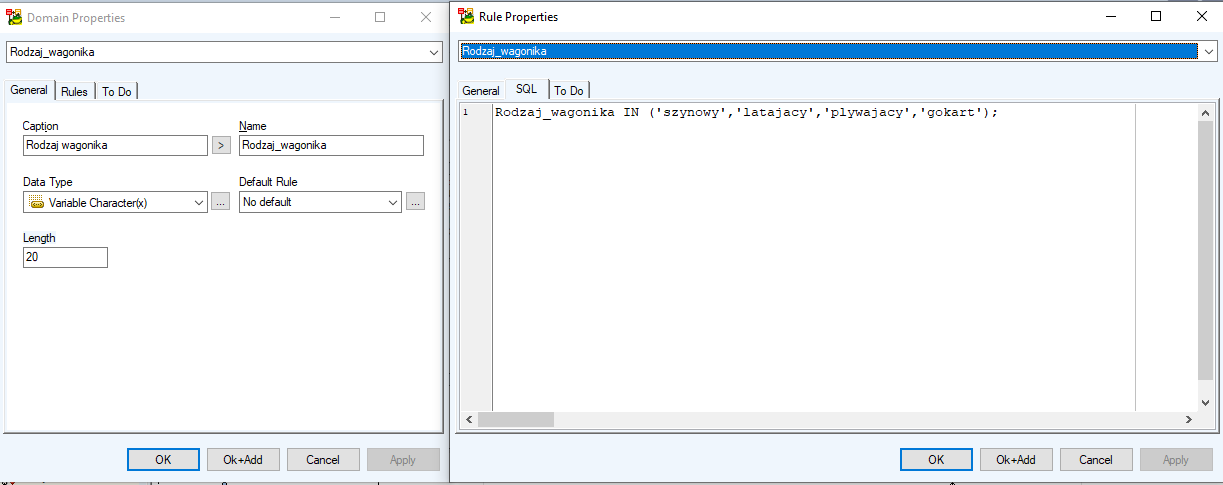
* Encja atrakcja:



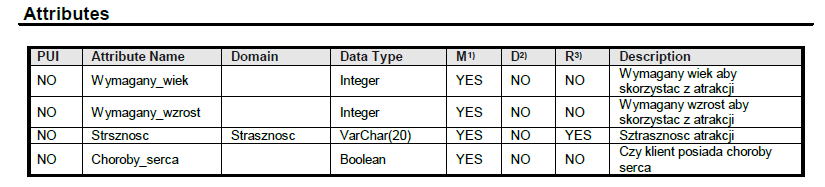


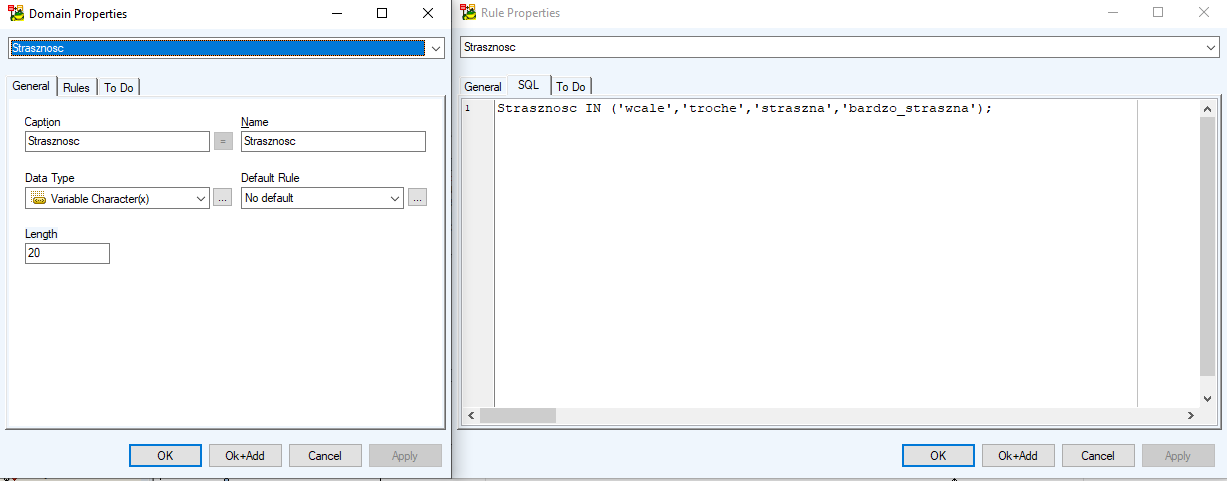
* Encja wagonik:



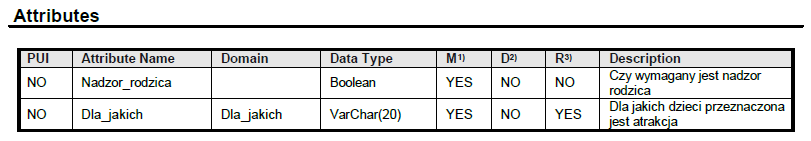


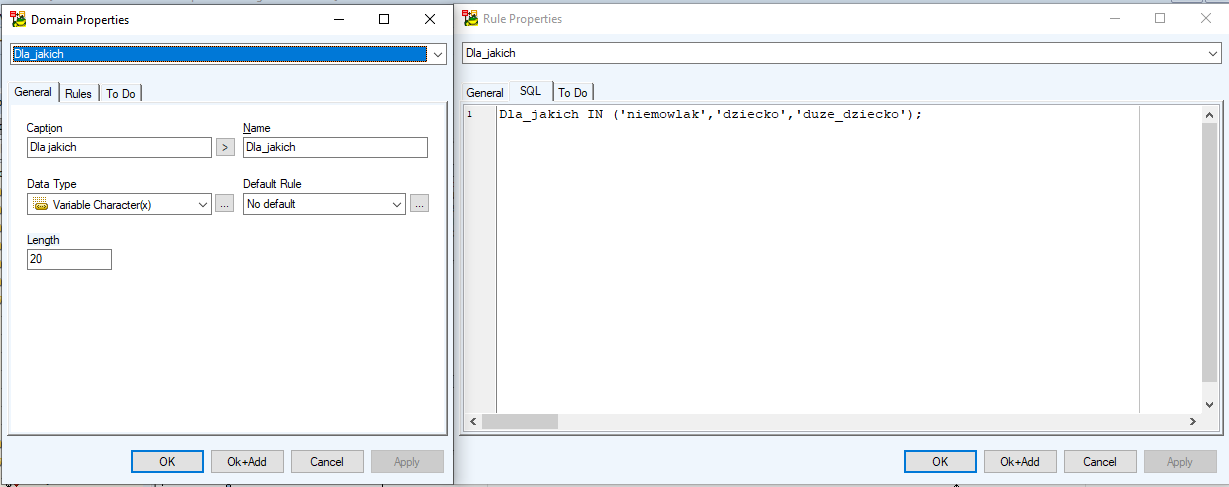
* Encja dla\_doroslych (child entity of Atrakcja):



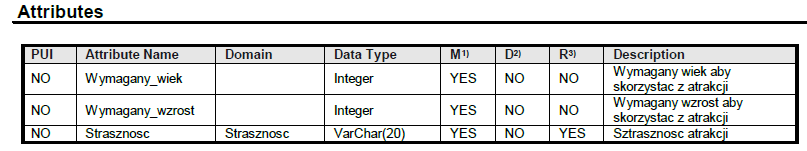


* Encja dla\_dzieci (child entity of Atrakcja):



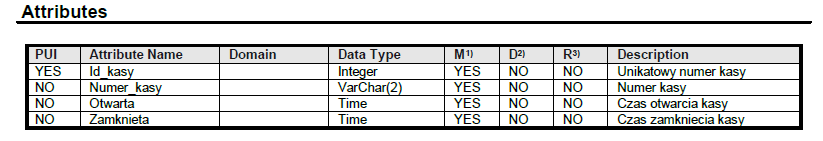


* Encja dla\_mlodziezy (child entity of Atrakcja):

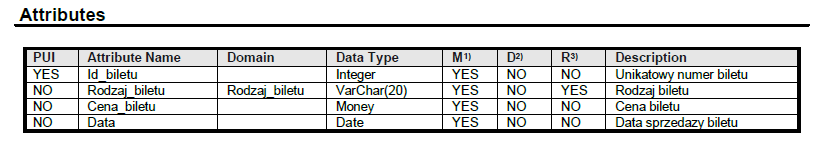


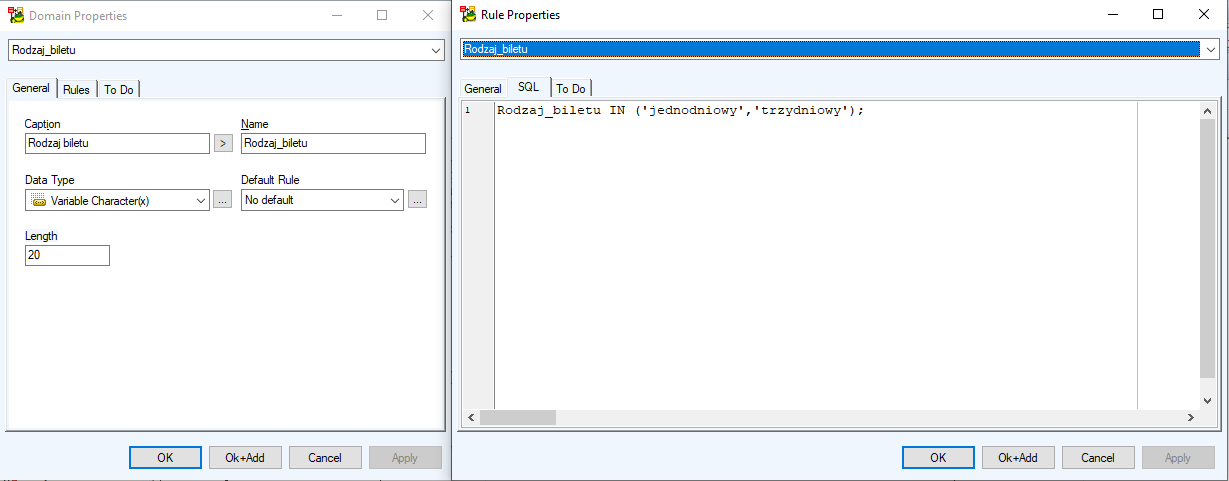
Domena i reguła odnosząca się do Strasznosc znajduje się wyżej (encja dla\_doroslych).

* Encja kasa:

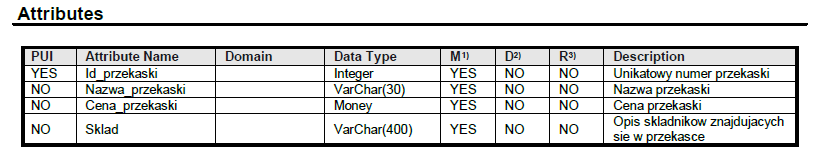


* Encja bilet:

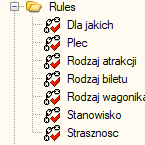
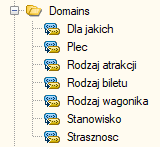




* Encja przekaska:



# 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.
* Strasznosc – reguła występująca w encji dla\_doroslych oraz dla\_mlodziezy. Określa straszność danej atrakcji, a jest ona mierzona w skali: wcale, trochę, straszna, bardzo straszna.

# 3.5 Klucze kandydujące i główne:

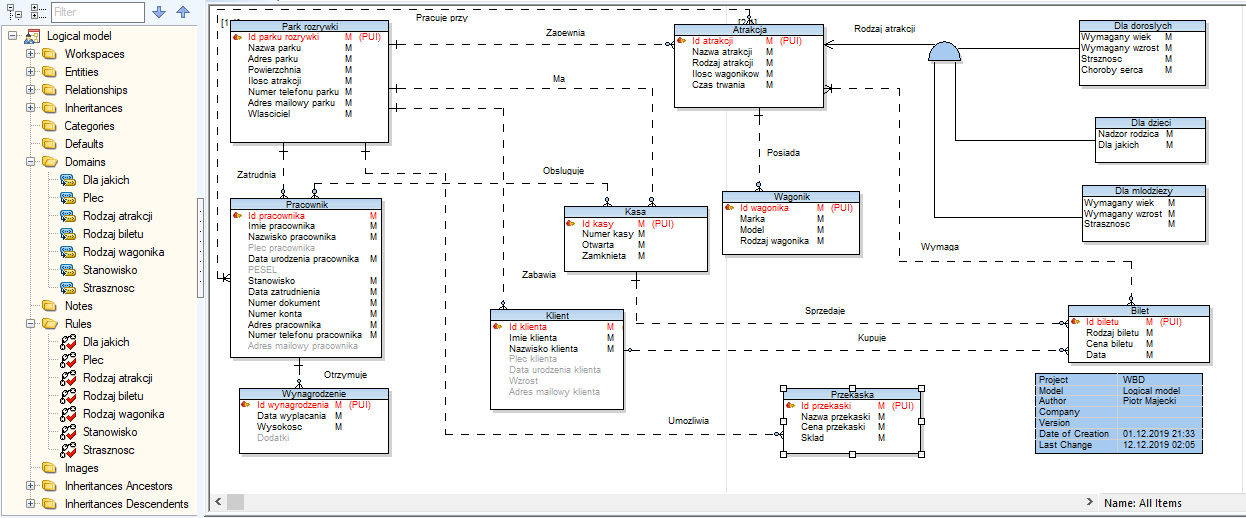
Aby uniknąć niejednoznaczności oraz błędów w Naszej bazie danych postanowiliśmy stworzyć sztuczne klucze główne, które identyfikujemy jako ID danej encji. Dzięki takiemu rozwiązaniu mamy gwarancję, że każdy Nasz klucz główny jest unikatowy, co daje Nam spójność bazy danych.

Jak widzimy niektóre encje nie mają w ogóle odpowiedniego klucza kandydującego. Jeżeli chodzi o encje 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 udałoby 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 placowka (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 oddzial (w danym oddziale pracowaliby pracownicy). Wtedy encja pracownik byłaby połączona związkiem z encją oddzial (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 oddzialem), 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 boolen. 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:

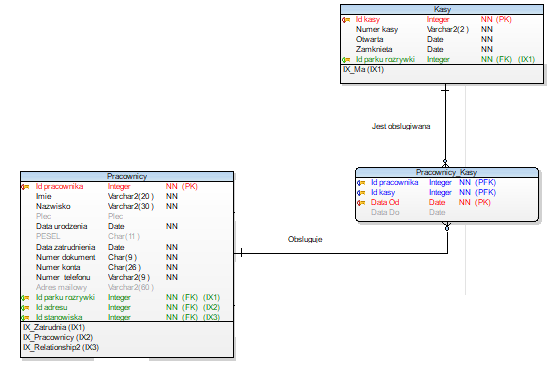
N1) - Not Null

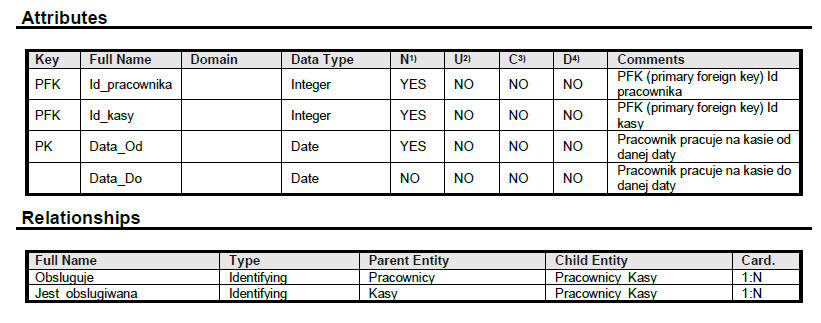
U2) - Unique

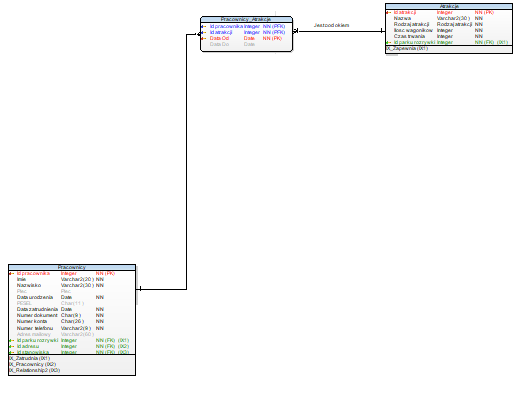
C3) - Check

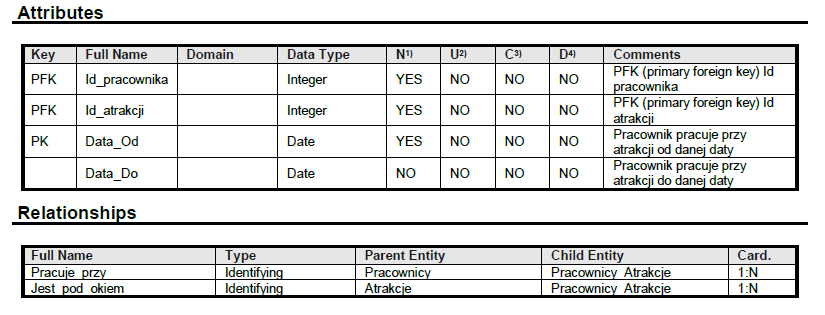
D4) - Default

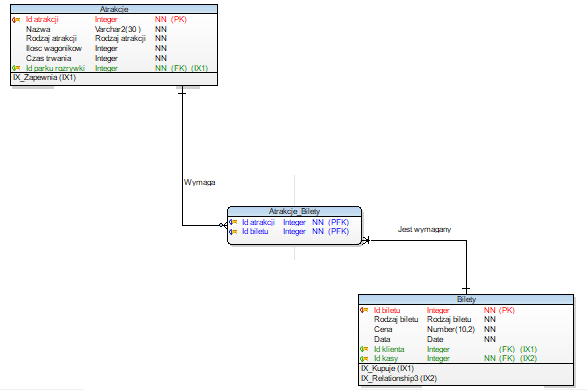
* Usunięcie związków wiele do wielu (M:N):



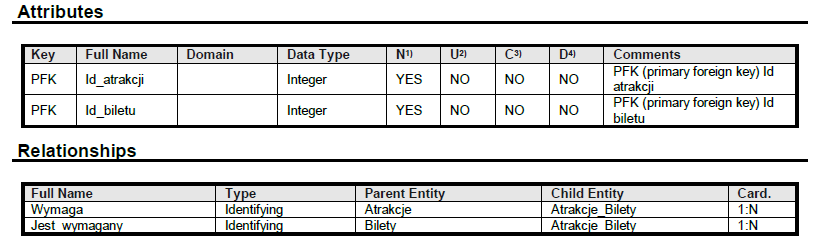
Zbiór encja Pracownicy\_Kasy:  




Zbiór encja Pracownicy\_Atrakcje:  




Zbiór encja Atrakcje\_Bilety:



* Zmiana nazwy encji:

Park rozrywki – Parki rozrywki

Pracownik – Pracownicy

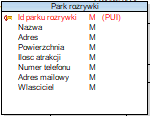
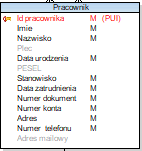
Wynagrodzenie – Wynagrodzenia  
Klient – Klienci

Kasa – Kasy

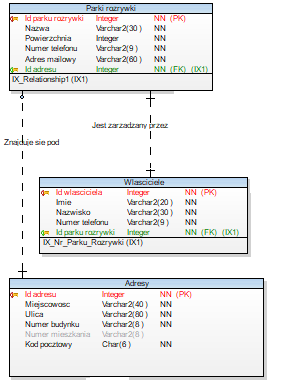
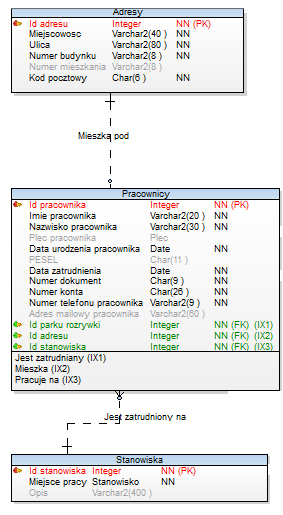
Itd.

* Usunięcie pól segmentowych:

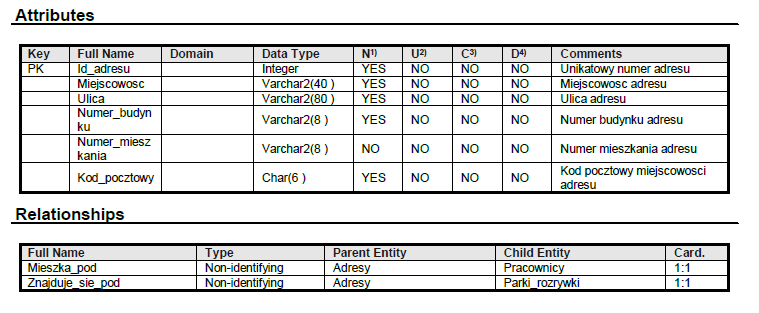
Model konceptualny:

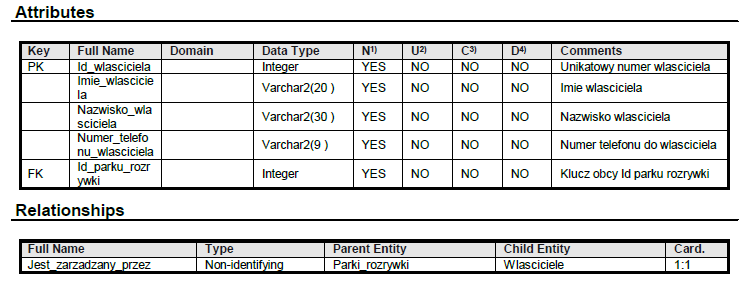
Model logiczny:

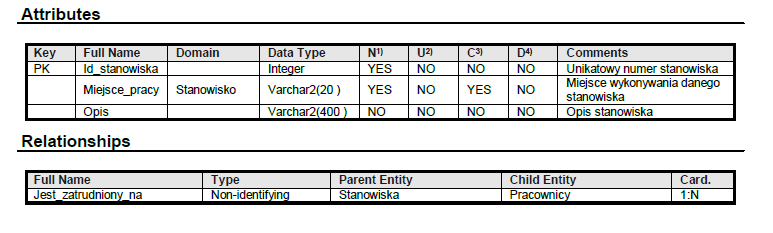
Zbiór encji Adresy:



Zbiór encji Wlasciciele:



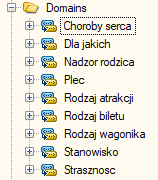
Zbiór encji Stanowiska:



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

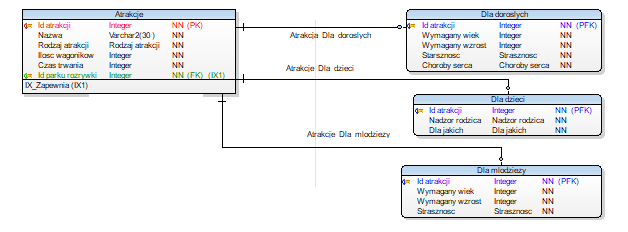
* Nowe dziedziny i reguły:

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

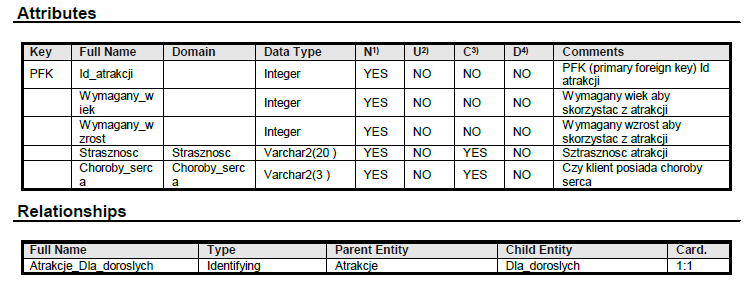


1. Choroby serca - reguła występująca w zbiorze encji dla\_doroslych. W modelu konceptualnym był to atrybut typu boolen. Określa czy klient ma problemy z sercem. Odpowiedzi to: ‘tak’ lub ‘nie’.
2. Nadzor rodzica - reguła występująca w zbiorze encji dla\_dzieci. W modelu konceptualnym był to atrybut typu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

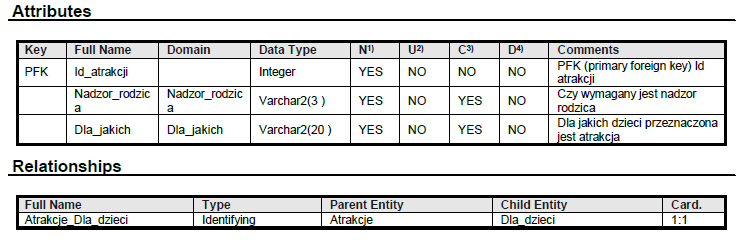
* Dziedziczenie:



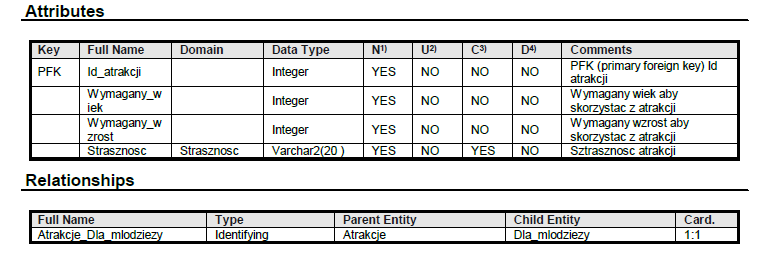
Zbiór encja Dla\_doroslych:



Zbiór encji Dla\_dzieci:



Zbiór encji Dla\_mlodziezy:



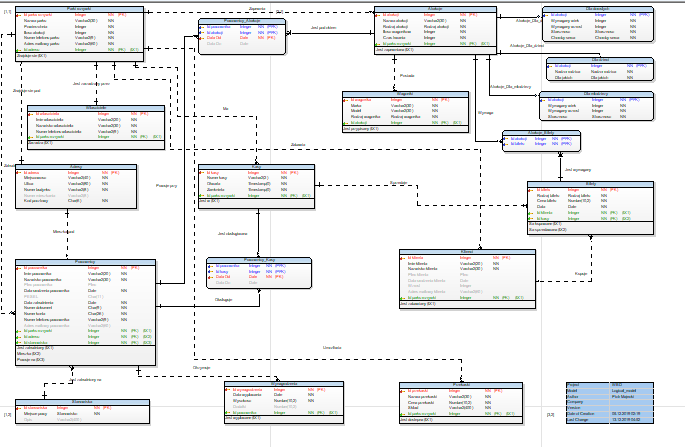
# 4.3 Proces normalizacji:

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:

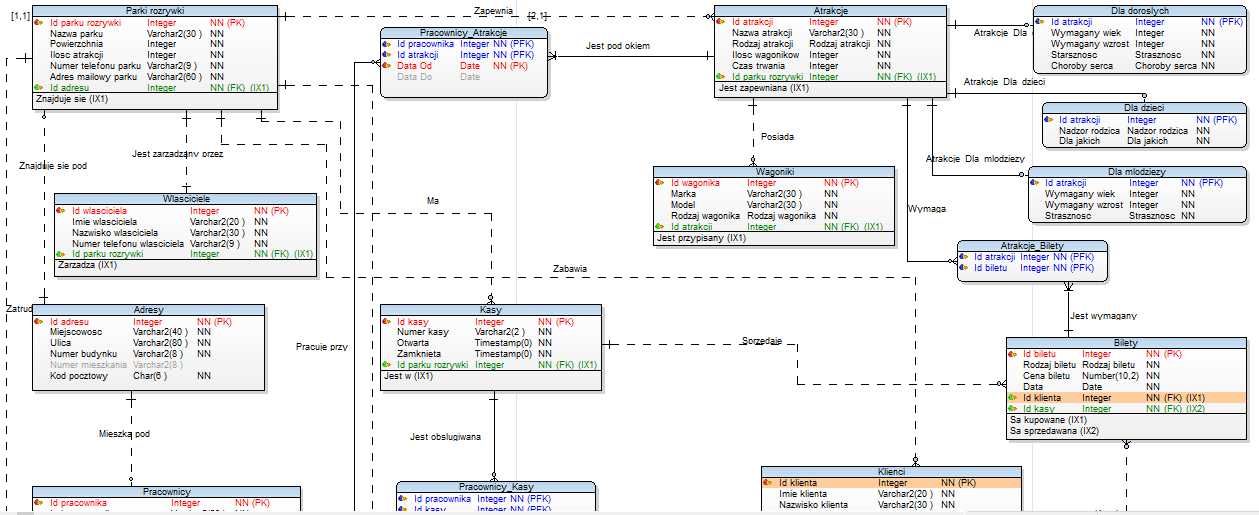
* Każda wartość atrybutu każdego zbioru encji jest wartością elementarną.
* Rozbicie Adresu na wartości atomowe (Miejscowosc, ulica, numer\_budynku, numer\_mieszkania, kod pocztowy).
* Utworzenie nowych zbiorów encji: Adresy, Wlasciciele, Stanowiska.
* Wszystkie atrybuty są unikatowe i mają inne nazwy.
* 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:

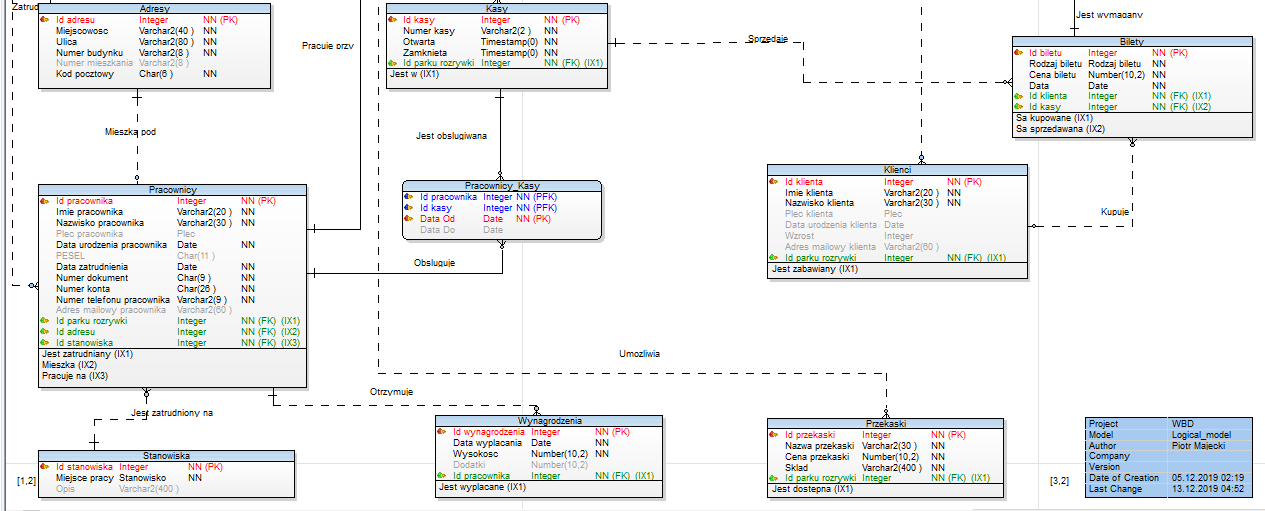
* Całość:



* Góra:



* Dół:



# 4.5 Więzy integralności:

W celu zabezpieczenia Naszej bazy danych przed błędnym dodaniem danych lub niewłaściwym modyfikowaniu 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).

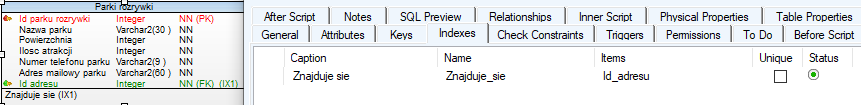
|  |  |  |
| --- | --- | --- |
| Numer | Transakcja/Funkcjonalność | Spełniona? |
| 1 | Modyfikacja danych parku rozrywki | TAK |
| 2 | Przegląd danych parku rozrywki | TAK |
| 3 | Dodawanie/Usuwanie pracowników | TAK |
| 4 | Modyfikacja danych pracownika | TAK |
| 5 | Przegląd danych pracownika | TAK |
| 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.

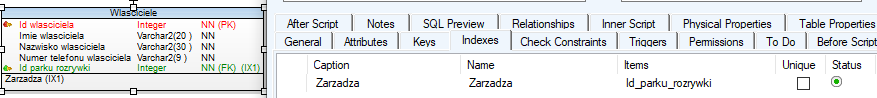
# 5.2 Strojenie bazy danych – dobór indeksów:

Stworzone indeksy:

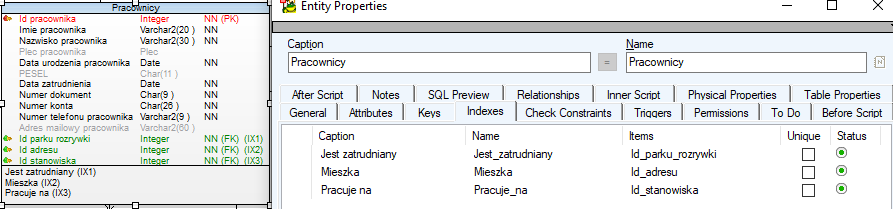
* Parki rozrywki:



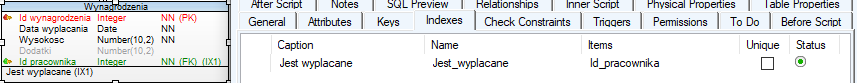
* Wlasciciele:



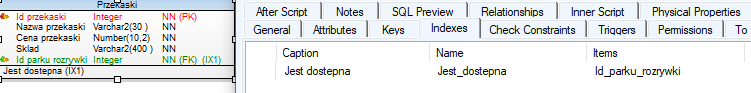
* Pracownicy:



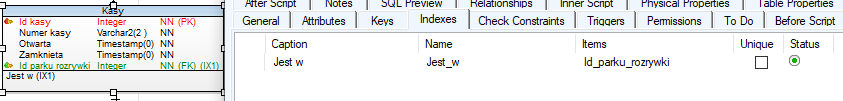
* Wynagrodzenia:



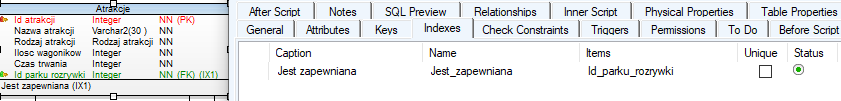
* Przekaski:



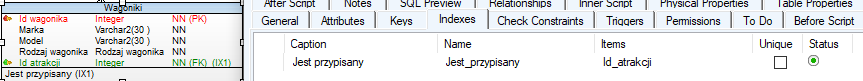
* Kasy:



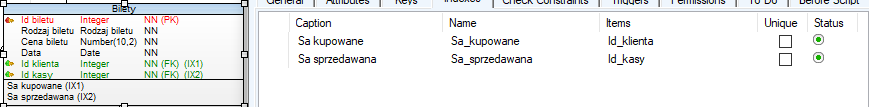
* Atrakcje:



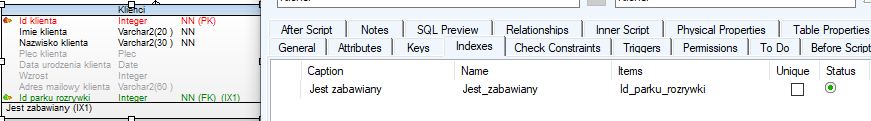
* Wagoniki:



* Bilety:



* Klienci:



# 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\_parku IS 'Numer telefonu do parku rozrywki';

COMMENT ON COLUMN Parki\_rozrywki.Adres\_mailowy\_parku 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\_zatrudniany 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\_pracownika 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\_pracownika IS 'Numer telefonu pracownika';

COMMENT ON COLUMN Pracownicy.Adres\_mailowy\_pracownika IS 'Adres e-mail pracownika';

COMMENT ON COLUMN Pracownicy.Id\_parku\_rozrywki IS 'Klucz obcy Id praku 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\_wyplacania 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\_wyplacane 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\_wyplacania IS 'Data wyplacania 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 klienia';

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\_mlodziezy','dla\_doroslych'));

-- 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','gokart'));

-- 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,

Strasznosc 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 Strasznosc CHECK (Strasznosc IN ('wcale','troche','straszna','bardzo\_straszna'));

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.Strasznosc IS 'Sztrasznosc 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,

Strasznosc 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 Strasznosc\_mlodzez CHECK (Strasznosc IN ('wcale','troche','straszna','bardzo\_straszna'));

-- 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\_dziecko'));

-- 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,

Zamknieta 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.Zamknieta IS 'Czas zamkniecia 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\_pracownika 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 Wlasciciele

CREATE TABLE Wlasciciele(

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 Wlasciciele

CREATE INDEX Zarzadza ON Wlasciciele (Id\_parku\_rozrywki);

-- Add keys for table Wlasciciele

ALTER TABLE Wlasciciele ADD CONSTRAINT PK\_Wlasciciele PRIMARY KEY (Id\_wlasciciela);

-- Table and Columns comments section

COMMENT ON COLUMN Wlasciciele.Id\_wlasciciela IS 'Unikatowy numer wlasciciela';

COMMENT ON COLUMN Wlasciciele.Imie\_wlasciciela IS 'Imie wlasciciela';

COMMENT ON COLUMN Wlasciciele.Nazwisko\_wlasciciela IS 'Nazwisko wlasciciela';

COMMENT ON COLUMN Wlasciciele.Numer\_telefonu\_wlasciciela IS 'Numer telefonu do wlasciciela';

COMMENT ON COLUMN Wlasciciele.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\_Stanowiska 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\_obslugiwana 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.

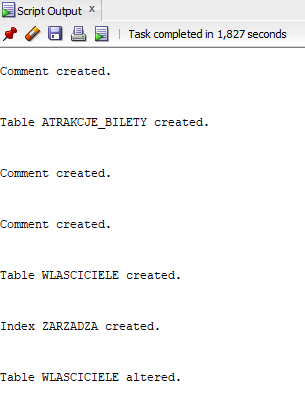
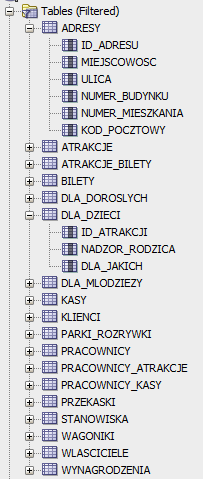
Wszystkie sekwencje się dodały.

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', '12345678912354678912345678', '777777777',' fajamarlena@costam.pl', 1, 8, 7);

INSERT INTO pracownicy VALUES (8, 'Filip', 'Kubek', 'M', '1985/05/18', NULL, '2018/09/19', 'NIG12A345', '12345678912345678912435678', '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, 'Ciepla bulka typ hot-dog z pyszna parowka z indyka i keczupem ', 1);

INSERT INTO przekaski VALUES (2, 'Hamburger', 12, 'Wypieczona duza bulka, z soczystym miesem wieprzowym, polana sosem 1000 wysp z dodatkiem pomidora, ogorka oraz prazonej cebulki', 1);

INSERT INTO przekaski VALUES (3, 'Zapiekanka', 9, '30cm bulka, z pieczarkami, szynka, serem zoltym, 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 mlot', 'dla\_doroslych', 1, 480, 1);

INSERT INTO atrakcje VALUES (2, 'Koniki', 'dla\_dzieci', 5, 300, 1);

INSERT INTO atrakcje VALUES (3, 'Dziki waz', 'dla\_mlodziezy', 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', 'Wiercipieta', '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\_mlodziezy 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');

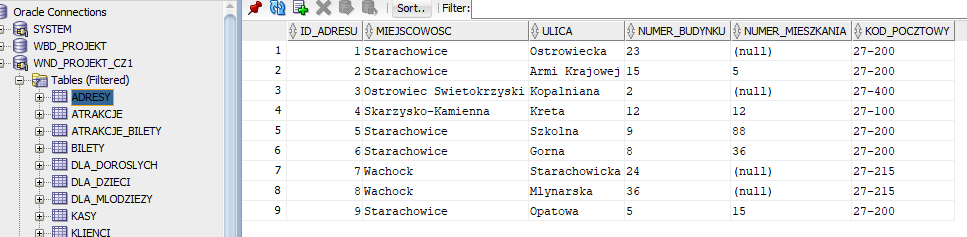
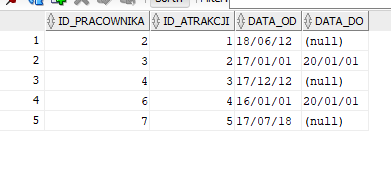
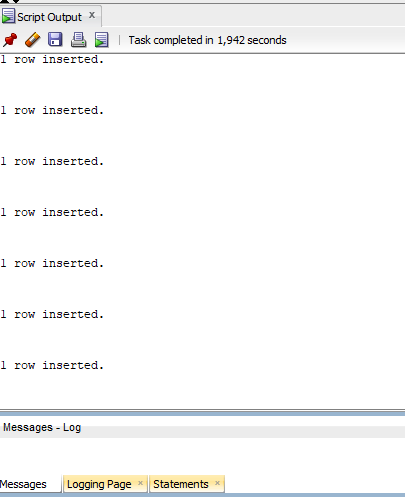
INSERT INTO pracownicy\_atrakcje VALUES (7, 5, '2017/07/18', NULL);

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.

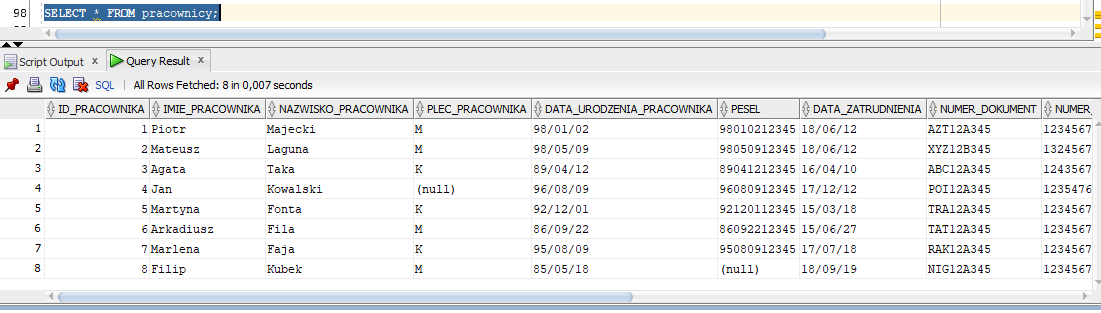
Wszystkie tabele się zaktualizowały i dane zostały dodane.



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

* Wypisz wszystko o pracownikach:

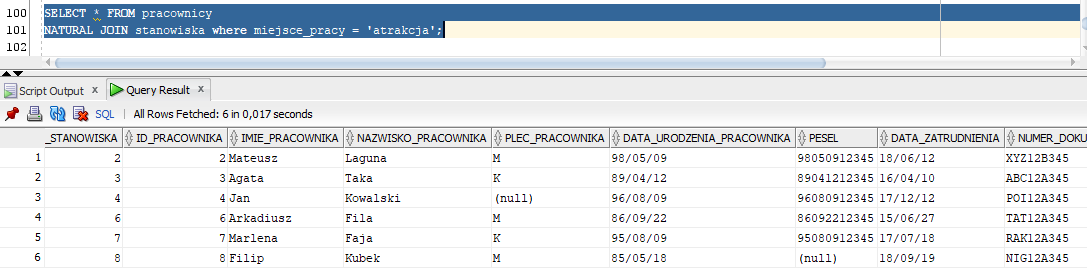
SELECT \* FROM pracownicy;



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

SELECT \* FROM pracownicy

NATURAL JOIN stanowiska where miejsce\_pracy = 'atrakcja';

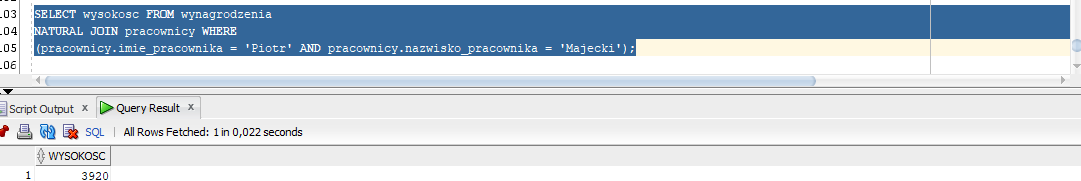


* Wypisz wysokość wynagrodzenia pracownika, którego imię to Piotr, a nazwisko to Majecki:

SELECT wysokosc FROM wynagrodzenia

NATURAL JOIN pracownicy WHERE

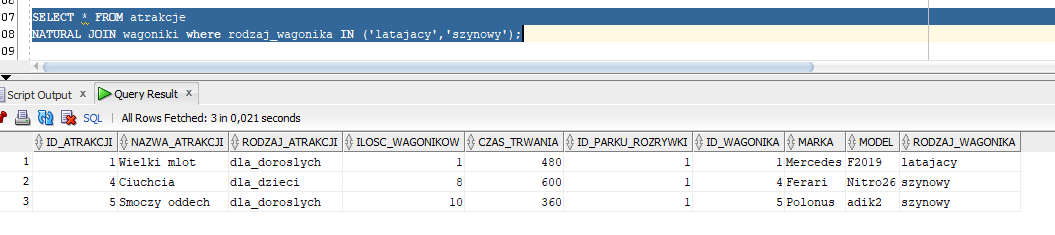
(pracownicy.imie\_pracownika = 'Piotr' AND pracownicy.nazwisko\_pracownika = 'Majecki');



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

SELECT \* FROM atrakcje

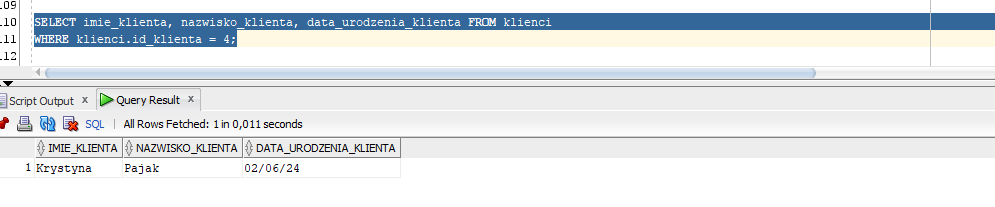
NATURAL JOIN wagoniki where rodzaj\_wagonika IN ('latajacy','szynowy');



* Wypisz takie dane jak imię klienta, nazwisko klienta, data urodzenia klienta dla klienta o Id\_klienta równym 4:

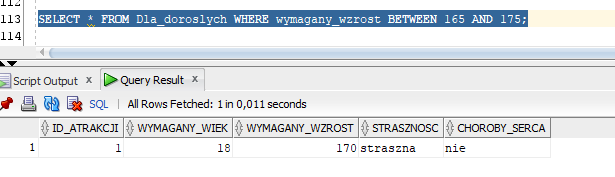
SELECT imie\_klienta, nazwisko\_klienta, data\_urodzenia\_klienta FROM klienci

WHERE klienci.id\_klienta = 4;



* 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;



# **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.