Zintegrowany Program Rozwoju Politechniki Lubelskiej - część druga

POLITECHNIKA LUBELSKA
WYDZIAŁ ELEKTROTECHNIKI I INFORMATYKI

KIERUNEK STUDIÓW INFORMATYKA

Przedmiot: Wprowadzenie do systemów baz danych

Raport z wykonania projektu pt.

System wspomagający funkcjonowaniu Rowera Miejskiego

Autory:
Darya Benedziktovich
Mykola Amelin
Viktoryia Barysevich

Lublin, 2021

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Społeczny

Rozdział 1. OPIS WYBRANEGO OBSZARU RZECZYWISTOŚCI ORAZ WSKAZANIE PROBLEMU, KTÓRY ZOSTANIE ROZWIĄZANY PRZY POMOCY SYSTEMU INFORMATYCZNEGO WYKORZYSTUJĄCEGO PROJEKTOWANĄ BAZĘ DANYCH

Rzeczywistość systemu wspomagającego funkcjonowaniu Rowera Miejskiego na samej sprawie nie jest bardzo skomplikowana. Sam system może obejmować wiele krajów i miast, obsługiwać tyśiący użytkowników i rowerów, ale być prostym oraz łatwym do ogarnięcia.

W rzeczywistości systemu rowerów miejskich możemy wyróżnić następujące obiekty:

- Użytkownik – osoba, korzystająca z systemu Roweru Miejskiego,
- Rower,
- Stacja Rowerową,
- Terminał Stacji Rowerowej (opłata przejazdu, wypożyczenie/swrot Rowerów),

Oraz procesy, realizowane w tym systemie:

- Rejestracja Użytkowników w systemie i wprowadzenie danych osobowych,
- Logowanie Użytkowników w systemie,
- Zmiana danych osobowych Użytkowników,
- Doładowanie konta Użytkowników,
- Przegląd Stacji Rowerowych i Rowerów,
- Rezerwacja Roweru,
- Przejazd;
- Zgłoszenie problemu.

Logiczna kolejność przedstawionych procesów będzie przedstawiona w rozdziale 3.

W wybraneo przez nas rzeczywistości możemy określić następujące problemy, które należy rozwiązać poprzez wdrożenie systemu informatycznego:

- Brak kontaktu między użytkownikami a osobami obsługującymi system,
- Brak możliwości tworzenia konta dla Użytkowników systemu,
  - Brak wygodnego sposobu opłaty i śledzenie podróży,
  - Brak możliwości przeglądu historii przejazdów,
- Brak możliwości sterowania stanów Rowerów oraz Stacji Rowerowych przez osoby obsługujące system,

Rozdział 2. SFORMUŁOWANIE CELU BUDOWY SYSTEMU INFORMATYCZNEGO, WYKORZYSTUJĄCEGO PROJEKTOWANĄ BAZĘ DANYCH, ORAZ PREZENTACJA WYMAGAŃ FUNKCJONALNYCH STAWIANYCH SYSTEMOWI

Celem budowy systemu informatycznego, który będzie wykorzystował zaprojektowaną przez nas bazę danych, jest ułatwienie obsługi Użytkowników oraz Rowerów i Stacji Rowerowych. Ten system będzię w stanie w łatwy sposób obsługiwać jak Użytkowników, zapewniajac szybkie działanie głownych możliwości systemu oraz terminową pomoc ze strony serwisu w razie jakichś problemów, tak i Rowery ze Stacjiami Rowerowymi, sterując ich stan oraz szybko reagują na awarie.

Poniżej są przedtawione wymagania funkcjonalne jakie powinny być spełnione przez projektowany system IT:

- System informatyczny jest przedstawiony jako aplikacja mobilna,
- Działający w czasie rzeczywistym 24/7 (czyli automatyczne działanie systemu),
- Zapewniający ubsługę wielu użytkowników jednocześnie,
- Zapewniający strerowanie przejazdów (długość przejazdów, trasę itd.),
- Zapewniający możliwość zgłoszenia problemów przez Użytkowników,
- Zapewniający możliwość gromadzenia historii wypożyczeń,
- Zapewniający możliwość poszukiwania Rowerów i Stacji Rowerowych konkretnych typów,
- Posada obsługę dynamicznej mapy Stacji w mieście,
- Każda Stacja Rowerowa będzie opatrzona o terminal, za pomocą którego Użytkowniki będą w stanie dokonać wypożyczenia oraz zwroty Rowerów itd.,
- Zarządca systemem musi ustawić minimalną kwotę pieniędżną (dalej ”opłata inicjalna”) którą musi posiadać każdy Użytkownik który chcę dokonywać przejazdy.

Rozdział 3. SZCZEGÓŁOWY OPIS PROCESÓW REALIZOWANYCH W WYBRANEJ RZECZYWISTOŚCI, ZALEŻNOŚCI MIĘDZY NIMI ORAZ OBIEKTÓW W NICH UCZESTNICZĄCYCH

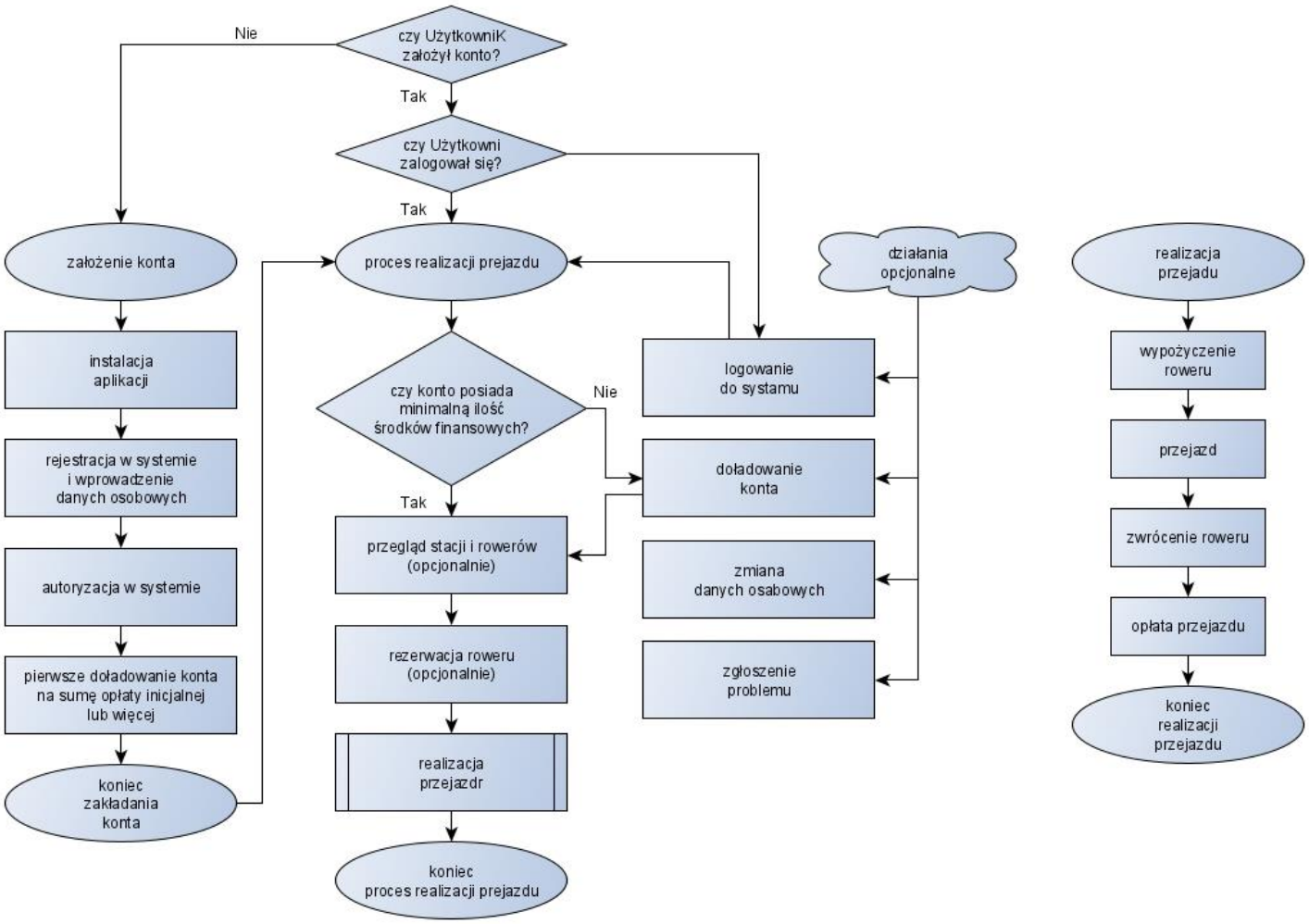
Tabela 3.1. Procesy jakie będą zachodzić w wybranej rzeczywistości

Proces	Obiekty uczestniczące	Opis
Założenie konta	Użytkownik	Po instalacji aplikacji Użytkownik musi 1) zarejestrować się w systemie (wprowadzić dane osobowe), 2) autoryzować się, 3) po raz pierwszy doładować konto na sumę opłaty inicjalnej lub więcej.
Logowanie do systemu	Użytkownik	Jeżeli Użytkownik już posiada konto to ma on możliwość zalogować się na innych urządzeniach wprowadzając dane osobowe.
Zmiana danych osobowych	Użytkownik	Użytkownik jest w stanie zmienić swoje dane osobowy (np. w przypadku zmiany miasta lub numeru telefona).
Doładowanie konta	Użytkownik	Użytkownik musi terminowo doładowywać konto. W tym celu 1) musi wybrać metodę płatności, 2) doładować konto wykorzystując wybraną metodę płatności. Na koncjie Użytkownik ma mięć przynajmniej ilość środków równą opłacie inicjalnej.
Przegląd stacji i rowerów	Użytkownik, Rower, Stacja Rowerowa	Użytkownik ma możliwość zobaczyć ilość Rowerów na poszczególnych Stacjach Rowerowych oraz jakie Rowery są na konkretnych Stacjach Rowerowych. Dla tych celej system informatyczny posiada dynamiczną mapę, na której są zaznaczone Stacje Rowerowe oraz ilość Rowerów na nich. Po przejściu do konkretnej Stacji Rowerowej można zobaczyć jakie Rowery są na tej stacji oraz przejść do szczegółów konkretnych Rowerów.
Rezerwacja roweru	Użytkownik, Rower	Użytkownik ma możliwość darmowej rezerwacji Roweru na określony przez system informatyczny czas. Dlatego trzeba 1) przejść do szczegółów dostępnego Rowera w aplikacji, 2) kliknąć ”Rezerwacja”

Proces	Obiekty uczestniczące	Opis
Realizacja przejazdu	Użytkownik, Rower, Stacja Rowerowa	Jest to skomplikowany proces, dlatego będzie on opisany w osobnej tabeli poniżej
Zgłoszenie problemu	Użytkownik	W razie jakichś problemów Użytkownik ma możliwość powiadomić o tym ludzi, którzy utrzymują system. Dlatego 1) Użytkownik musi w aplikacji przejść do zakładki ”Zgłośic problem”, 2) wybrać kategorię problemu (problem z wypożyczeniem roweru, zgłośić zepsuty problem itd.), 3) szczegółowo opisać zaistniały problem, 4) Naciśnąć przycisk ”Powiadomić”

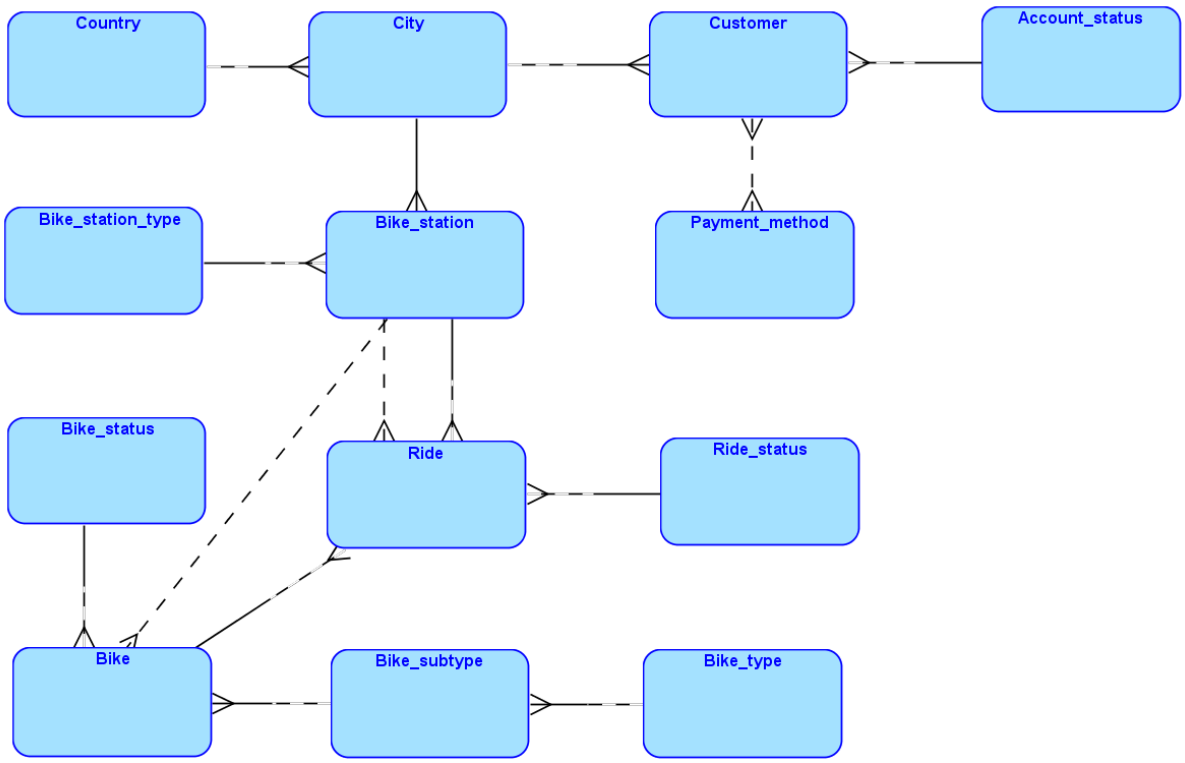
Tabela 3.2. Szczegółowy opis procesu "Realizacja przejazdu"

Podproces	Obiekty uczestniczące	Opis
Wypożyczanie roweru	Użytkownik, Rower, Stacja Rowerowa	Ten proces składa się z kilku etapów: 1) Wybór Stacji Rowerowej, 2) Wybór Roweru na tej Stacji Rowerowej, 3) Dokonanie wypożyczenia (wybierz ”Wypożycz rower” w aplikacji); Po wykonaniu tego kroku otrzymasz kod do zamka szyfrowego na rowerze, 4) Jeżeli jest to potrzebne, odblokuj Rower za pomocą otrzymanego kodu do zamka szyfrowego, 5) Odbirz zwolniony Rower ze Stacji Rowerowej Wynikiem tego podprocesu jest początek przejazdu
Przejazd	Użytkownik, Rower	To jest proces pośredni pomiędzy wypożyczeniem a zwróceniem Roweru, kiedy Użytkownik robi swój przejazd.
Zwrócenie roweru	Użytkownik, Rower, Stacja Rowerowa	Żeby zwrócić Rower na Stanje Rowerową Użytkownik musi wprowadzić Rower do elektrozamka oraz doczekać się sygnału dźwiękowego. W tym przypadku Użytkownik nie korzysta z systemu informatycznego. Jednak jeśli na Stacji Rowerowej nie ma wolnego elektrozamka, Użytkownik musi przypninąć Rower obejmą do ucha stojaka lub innego prawidłowo zabezpieczonego Roweru, a następnie wybrać opcję ”Zwrot” w aplikacji.
Oplata przejazdu	Użytkownik	Pieniądze są automatyczne wypłacane z konta Użytkownika zgodnie z czasem podróży.



Rys. 3.1. Zależności pomiędzy poszczególnymi procesami w logicznej kolejności ich realizacji

Rozdział 4. MODEL KONCEPTUALNY PROJEKTOWANEJ BAZY DANYCH



Rys. 4.1. Model konceptualny projektowanej bazy danych w formie graficznej.

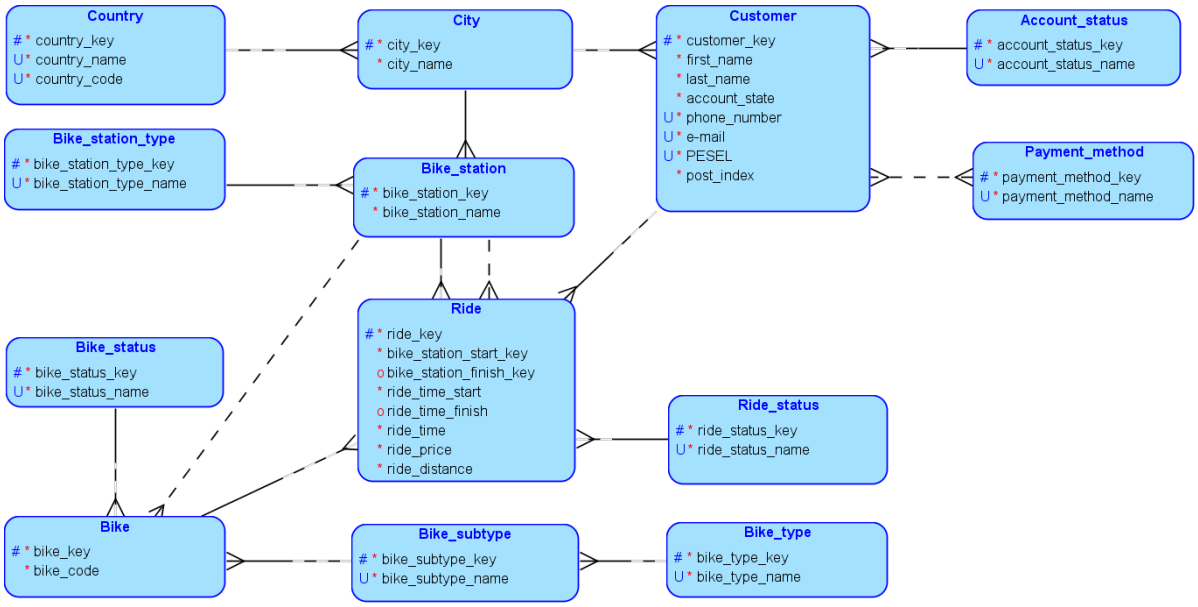
Tabela 4.1. Tabela opisująca encje

Nazwa encji	**Opis encji**
Country	Informacje o kraju, obejmujące nazwę państwa i kod kraju
City	Informacje o mieście, obejmujące nazwę miasta
Customer	Informacje o użytkowniku, posiadającym konto w systemie informatycznym, umożliwiającym wypożyczanie rowerów, obejmujące imię, nazwisko, ilość zgromadzonych środków finansowych, numer telefonu, email, PESEL i kod pocztowy użytkownika
Account_status	Informacje o stanie konta Użytkownika, obejmujące nazwę stanu
Payment_method	Informacje o metodzie płatności, dostępnych na platformie, obejmujące nazwę metody
Ride	Informacje o przejeździe, obejmujące numer Stacji początkowej oraz końcowej, czas rozpoczęcia i zakończenia przejazdu, czas trwania przejazdu oraz cenę
Ride_status	Informacje o stanie przejazdu, obejmujące nazwę stanu
Bike_station	Informacje o Stacji Rowerowej, obejmujące nazwę stacji oraz jej numer
Bike_station_type	Informacje o typie Stacji Rowerowej, obejmujące nazwę typu Stacji Rowerowej
Bike_type	Informacje o typie Roweru, obejmujące nazwę typu
Bike_subtype	Informacje o modelu Roweru, obejmujące nazwę podtypu
Bike	Informacje o Rowerze, obejmujące numer roweru
Bike_status	Informacje o stanie Roweru, obejmujące nazwę stanu

Tabela 4.2. Rodzaje związków pomiędzy poszczególnymi encjami

Związek	**Rodzaj**
Country -> City	Związek 1:N
City -> Customer	Związek 1:N
Account_status -> Customer	Związek 1:N
Payment_method -> Customer	Związek N:N
Customer -> Ride	Związek 1:N
Ride_status -> Ride	Związek 1:N
Bike_type -> Bike_subtype	Związek 1:N
Bike_subtype -> Bike	Związek 1:N
Bike_status -> Bike	Związek 1:N
City -> Bike_station	Związek 1:N
Bike_station_type -> Bike_station	Związek 1:N
Bike_station -> Bike	Związek 1:N
Bike_station -> Ride	Związek 1:N
Bike -> Ride	Związek 1:N

Rozdział 5. MODEL ZWIĄZKÓW ENCJI PROJEKTOWANEJ BAZY DANYCH



Rys. 5.1. Model logiczny projektowanej bazy danych.

5.1. Znaczenie atrybutów dla poszczególnych encji występujących w prezentowanym modelu logicznym bazy danych

Tabela 5.1. Specyfikacja atrybutów encji Country

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
country_key	Liczba całkowita, max 99	Tak	Tak	Identyfikator kraju, umożliwiający jego rozróżnienie wśród wielu innych krajów. Identyfikator sztuczny.
country_name	Znakowy, max 50 znaków	Tak	Tak	Nazwa kraju, który jest w systemie. Deskryptor.
country_code	Znakowy, max 3 znaka	Tak	Tak	Unikatowy kod kraju (krótki identyfikator) Deskryptor.

Tabela 5.2. Specyfikacja atrybutów encji City

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
city_key	Liczba całkowita, max 999	Tak	Tak	Identyfikator miasta, umożliwiający jego rozróżnienie wśród wielu miast krajów. Identyfikator sztuczny.
city_name	Znakowy, max 50 znaków	Tak	Nie	Nazwa miasta, przypisanego do konkretnego państwa. Deskryptor.

Tabela 5.3. Specyfikacja atrybutów encji Customer

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
customer_key	Liczba całkowita, max 999999999	Tak	Tak	Identyfikator Użytkownika, umożliwiający jego rozróżnienie wśród wielu innych Użytkowników. Identyfikator sztuczny
first_name	Znakowy, max 30 znaków	Tak	Nie	Imię Użytkownika w systemie. Deskryptor.
last_name	Znakowy, max 50 znaków	Tak	Nie	Nazwisko Użytkownika w systemie. Deskryptor.
account_state	Liczba rzeczywista, max. 999,99	Tak	Nie	Ilość środków finansowych na koncie Użytkownika. Deskryptor.
phone_number	Znakowy, max 9 znaków	Tak	Tak	Numer telefonu używany przez Użytkownika podczas logowania się do systemie. Deskryptor.
e-mail	Znakowy, max 50 znaków	Tak	Tak	Adres mailowy używany przez Użytkownika podczas logowania się do systemie. Deskryptor.
PESEL	Znakowy, max 11 znaków	Tak	Tak	PESEL Użytkownika używany podczas rejestracji się w systemie. Deskryptor.
post_index	Znakowy, max 6 znaków	Tak	Nie	Indeks pocztowy używany przez Użytkownika podczas logowania się do systemie. Deskryptor.

Tabela 5.4. Specyfikacja atrybutów encji Payment_method

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
payment_method_key	Liczba całkowita, max 9	Tak	Tak	Identyfikator metody płatności, umożliwiający jego rozróżnienie wśród wielu innych metod płatności. Identyfikator sztuczny
payment_method_name	Znakowy, max 20 znaków	Tak	Tak	Nazwa metody płatności jakie wykorzystywanej w systemie. Deskryptor.

Tabela 5.5. Specyfikacja atrybutów encji Account_status

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
account_status_key	Liczba całkowita, max 9	Tak	Tak	Identyfikator stanu konta Użytkownika, umożliwiający jego rozróżnienie wśród wielu innych stanów konta. Identyfikator sztuczny
account_status_name	Znakowy, max 10 znaków	Tak	Tak	Nazwa stanu konta Użytkownika w systemie. Deskryptor.

Tabela 5.6. Specyfikacja atrybutów encji Bike_type

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
bike_type_key	Liczba całkowita, max 9	Tak	Tak	Identyfikator rodzaju Roware, umożliwiający jego rozróżnienie wśród wielu rodzajów Rowerów. Identyfikator sztuczny
bike_type_name	Znakowy, max 15 znaków	Tak	Tak	Nazwa typu Roweru. Deskryptor.

Tabela 5.7. Specyfikacja atrybutów encji Bike_subtype

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
bike_subtype_key	Liczba całkowita, max 99	Tak	Tak	Identyfikator modelu Roweru, umożliwiający jego rozróżnienie wśród wielu innych modeli Rowerów. Identyfikator sztuczny
bike_subtype_name	Znakowy, max 20 znaków	Tak	Tak	Nazwa modelu Rowera. Deskryptor.

Tabela 5.8. Specyfikacja atrybutów encji Bike

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
bike_key	Liczba całkowita, max 99999	Tak	Tak	Identyfikator Roweru, umożliwiający jego rozróżnienie wśród wielu innych Rowerów. Identyfikator sztuczny.
bike_code	Znakowy, max 4 znaka	Tak	Nie	Kodu do zamka szyfrowego Rowera. Deskryptor.

Tabela 5.9. Specyfikacja atrybutów encji Bike_status

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
bike_status_key	Liczba całkowita, max 9	Tak	Tak	Identyfikator stanu Rowera, umożliwiający jego rozróżnienie wśród wielu innych stanów Rowerów. Identyfikator sztuczny
bike_status_name	Znakowy, max 15 znaków	Tak	Tak	Nazwa stanu przejazdu. Deskryptor.

Tabela 5.10. Specyfikacja atrybutów encji Bike_station

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
bike_station_key	Liczba całkowita, max 9999	Tak	Tak	Identyfikator Stacji Rowerowej, umożliwiający jego rozróżnienie wśród wielu innych Stacji Rowerowych. Identyfikator sztuczny
bike_station_name	Znakowy, max 20 znaków	Tak	Nie	Nazwa Stacji Rowerowej. Deskryptor.

Tabela 5.11. Specyfikacja atrybutów encji Bike_station_type

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
bike_station_type_key	Liczba całkowita, max 9	Tak	Tak	Identyfikator rodzaju Stacji Rowerowej, umożliwiający jego rozróżnienie wśród innych rodzjajów Stacji Rowerowych. Identyfikator sztuczny
bike_station_type _name	Znakowy, max 15 znaków	Tak	Tak	Nazwa typu Stacji Rowerowej. Deskryptor.

Tabela 5.12. Specyfikacja atrybutów encji Ride

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
ride_key	Liczba całkowita, max 999999999	Tak	Tak	Identyfikator przejazdu, umożliwiający jego rozróżnienie wśród wielu innych przejazdów. Identyfikator sztuczny
bike_station_ start_key	Liczba całkowita, max 9999	Tak	Nie	Identyfikator stacji, na której został wypożyczony Rower. Deskryptor.
bike_station_ finish_key	Liczba całkowita, max 9999	Nie	Nie	Identyfikator stacji, do której został zwrócony Rower. Deskryptor.
ride_time_start	Data i godzina	Tak	Nie	Czas rozpoczęcia przejazdu. Deskryptor.
ride_time_finish	Data i godzina	Nie	Nie	Czas zakończenia przejazdu. Deskryptor.
ride_time	Liczba całkowita	Tak	Nie	Czas trwania przejazdu w sekundach. Deskryptor.
ride_price	Liczba całkowita, max 999999999	Tak	Nie	Cena przejazdu. Deskryptor.
ride_distance	Liczba rzezywista, max 99,9	Tak	Nie	Dystancja przejazdu w kilometrach. Deskryptor.

Tabela 5.13. Specyfikacja atrybutów encji Ride_status

Nazwa atrybutu	Typ i maksymalny rozmiar danych	Wartość wymagana (Tak/Nie)	Wartość unikatowa (Tak/Nie)	Opis atrybutu
ride_status_key	Liczba całkowita, max 9	Tak	Tak	Identyfikator stanu przejazdu. Identyfikator sztuczny.
ride_status_name	Znakowy, max 15 znaków	Tak	Tak	Nazwa stanu przejazdu. Deskryptor.

5.2. Istnienie związku pomiędzy poszczególnymi encjami (‘+’ oznacza istnienie)

Tabela 5.14. Istnienie związku pomiędzy poszczególnymi encjami

Nazwa encji	Country	City	Customer	Account_ status	Payment_ method	Bike_station _type	Bike_ station	Ride_ status	Ride	Bike_ status	Bike_ type	Bike_ subtype	Bike
Country		+											
City	+		+				+						
Customer		+		+	+				+				
Account_status			+										
Payment_method			+										
Bike_station_type							+						
Bike_station		+				+			+				+
Ride_status									+				
Ride			+				+	+					+
Bike_status													+
Bike_type												+	
Bike_subtype											+		+
Bike							+		+	+		+	

5.3. Szczegółowy opis każdego związku pomiędzy poszczególnymi encjami

Tabela 5.15. Opis związku między encjami Country i City

Tytuł	Opis
Nazwy encji	Country -> City
Opis	Państwo może zawierać miasto. Państwo może zawierać wiele miast. Nie każde państwo zawiera miasto. Dane miasto musi zawierać się w jakimś państwie. Dane miasto zawiera się tylko w jednym państwie.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.16. Opis związku między encjami City i Customer

Tytuł	Opis
Nazwy encji	City -> Customer
Opis	Miasto może odnosić się do Użytkownika. Miasto może odnosić się do wielu Użytkowników. Nie każde miasto odnosi się do Użytkowników. Dany Użytkownik musi odnosić się do jakiegoś miasta. Dany Użytkownik odnosi się tylko do jednego miasta.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.17. Opis związku między encjami Account_status i Customer

Tytuł	Opis
Nazwy encji	Account_status -> Customer
Opis	Stan konta może odnosić się do wielu Użytkowników. Nie każdy stan konta odnosi się do Użytkownika. Dany Użytkownik musi mieć jakiś stan konta. Dany Użytkownik ma tylko jeden stan konta.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.18. Opis związku między encjami Payment_method i Customer

Tytuł	**Opis**
Nazwy encji	Payment_method -> Customer
Opis	Metody płatności mogą odnosić się do wielu Użytkowników. Nie każda metoda płatności odnosi się do Użytkownika. Dany Użytkownik może mieć jakąś metodę płatności. Dany Użytkownik może mieć wiele metod płatności.
Stopień	Związek binarny
Typ	Związek N:N
Istnienie	Związek nieobowiązkowy

Tabela 5.19. Opis związku między encjami Bike_type i Bike_sybtype

Tytuł	**Opis**
Nazwy encji	Bike_type -> Bike_subtype
Opis	Typ Roweru może zawierać podtyp Roweru. Typ Roweru może zawierać wiele podtypów Roweru. Nie każdy typ Roweru zawiera podtyp Roweru. Dany podtyp Roweru musi zawierać się w jakimś typie Roweru. Dany podtyp Roweru zawiera się tylko w jednym typie Roweru.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.20. Opis związku między encjami Bike_sybtype i Bike

Tytuł	**Opis**
Nazwy encji	Bike_subtype -> Bike
Opis	Podtyp Roweru może zawierać Rower. Podtyp Roweru może zawierać wiele Rowerów. Nie każdy podtyp Roweru zawiera Rower. Dany Rower musi zawierać się w jakimś podtypie. Dany Rower zawiera się tylko w jednym podtypie.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.21. Opis związku między encjami Bike_status i Bike

Tytuł	**Opis**
Nazwy encji	Bike_status -> Bike
Opis	Stan Roweru może odnosić się do Roweru. Stan Roweru może odnosić się do wielu Rowerów. Nie każdy stan Roweru odnosi się do Roweru. Dany Rower musi mieć jakiś stan Roweru. Dany Rower ma tylko jeden stan Roweru.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.22. Opis związku między encjami Bike_station_type i Bike_station

Tytuł	**Opis**
Nazwy encji	Bike_station_type -> Bike_station
Opis	Typ Stacji Rowerowej może odnosić się do Stacji Rowerowej. Typ Stacji Rowerowej może odnosić się do wielu Stacji Rowerowych. Nie każdy typ Stacji Rowerowej odnosi się do Stacji Rowerowej. Dana Stacja Rowerowa musi odnosić się do jakiegoś typu Stacji Rowerowych. Dana Stacja Rowerowa odnosi się tylko do jednego typu Stacji Rowerowych.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.23. Opis związku między encjami Bike_station i Bike

Tytuł	**Opis**
Nazwy encji	Bike_station -> Bike
Opis	Stacja Rowerowa może zawierać Rower. Stacja Rowerowa może zawierać wiele Rowerów. Nie każda Stacja Rowerowa zawiera Rowery. Dany Rower może znajdować się na jakieś Stacji Rowerowej. Dany Rower może znajdować się tylko na jednej Stacji Rowerowej.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek nieobowiązkowy

Tabela 5.24. Opis związku między encjami City i Bike_Station

Tytuł	**Opis**
Nazwy encji	City -> Bike_station
Opis	Miasto musi zawierać Stację Rowerową. Miasto może zawierać wiele Stacji Rowerowych. Dana Stacja Rowerowa musi odnosić się do jakiegoś miasta. Dana Stacja Rowerowa odnosi się tylko do jednego miasta.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek obowiązkowy

Tabela 5.25. Opis związku między encjami Bike i Ride

Tytuł	**Opis**
Nazwy encji	Bike -> Ride
Opis	Rower może odnosić się do przejazdu. Rower może odnosić się do wielu przejazdów. Nie każdy Rower odnosi się do przejazdu. Dany przejazd musi mieć jakiś Rower. Dany przejazd ma tylko jeden Rower.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.26. Opis związku między encjami Customer i Ride

Tytuł	**Opis**
Nazwy encji	Customer -> Ride
Opis	Użytkownik może dokonać przejazdu. Użytkownik może mieć wiele przejazdów. Nie każdy Użytkownik ma przejazdy. Dany przejazd musi być dokonany przez jakiegoś Użytkownika. Dany przejazd został dokonany tylko przez jednego Użytkownika.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Tabela 5.27. Opis związku między encjami Bike_station_start i Ride

Tytuł	**Opis**
Nazwy encji	Bike_station_start -> Ride
Opis	Stacja Rowerowa może być punktem startowym w jakimś przejeździe. Stacja Rowerowa może być punktem startowym w wielu przejazdach. Przejazd musi zawierać stację początkową. Przejazd zawiera dokładnie jedną stację początkową.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

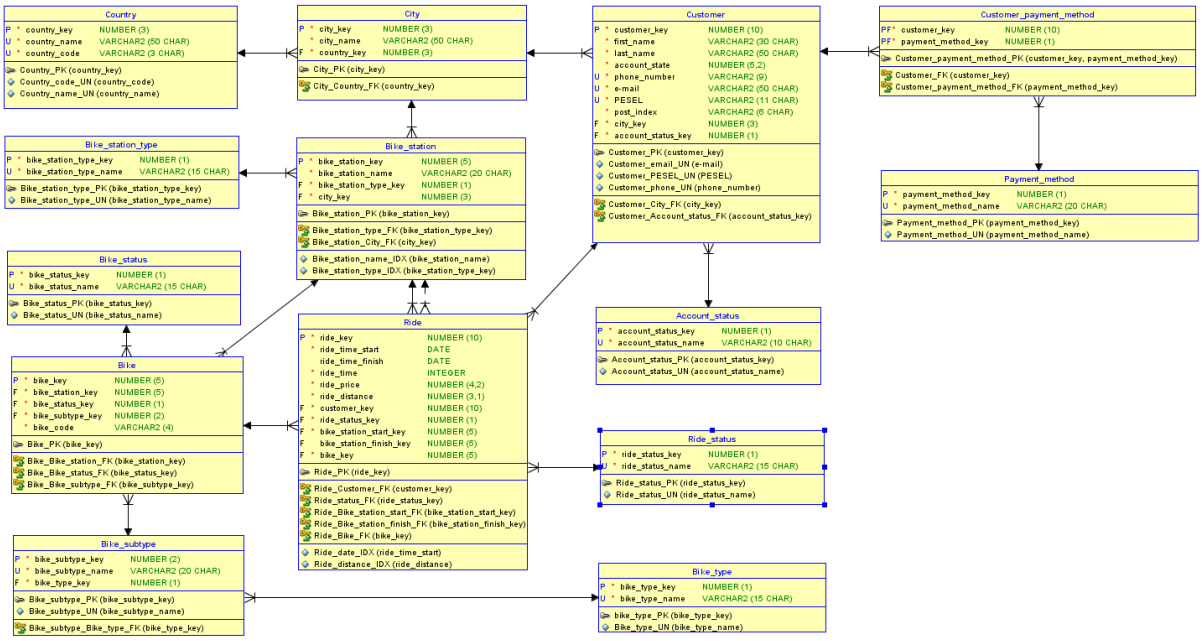
Tabela 5.28. Opis związku między encjami Bike_station_start i Ride

Tytuł	**Opis**
Nazwy encji	Bike_station_finish -> Ride
Opis	Stacja Rowerowa może być punktem końcowym w jakimś przejeździe. Stacja Rowerowa może być punktem końcowym w wielu przejazdach. Przejazd musi zawierać stację końcową. Przejazd może zawierać dokładnie jedną stację końcową.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek nieobowiązkowy

Tabela 5.29. Opis związku między encjami Ride_status i Ride

Tytuł	**Opis**
Nazwy encji	Ride_status -> Ride
Opis	Stan przejazdu może odnosić się do przejazdu. Stan przejazdu może odnosić się do wielu przejazdów. Nie każdy stan przejazdu odnosi się do przejazdu. Dany przejazd musi mieć jakiś stan przejazdu. Dany przejazd ma tylko jeden stan przejazdu.
Stopień	Związek binarny
Typ	Związek 1:N
Istnienie	Związek opcjonalny

Rozdział 6. MODEL RELACYJNY PROJEKTOWANEJ BAZY DANYCH



Rys. 6.1. Model relacyjny projektowanej bazy danych w formie graficznej

6.1. Struktury tabel

Tabela 6.1. Struktura tabeli Country

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
country_key	NUMBER (3)	TAK
country_name	VARCHAR2(50 CHAR)	TAK
country_code	VARCHAR2(3 CHAR)	TAK

Tabela 6.2. Struktura tabeli City

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
city_key	NUMBER (10)	TAK
city_name	VARCHAR2 (50 CHAR)	TAK
country_key	NUMBER (3)	TAK

Tabela 6.3. Struktura tabeli Account_status

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
account_status _key	NUMBER(1)	TAK
account_ status_name	VARCHAR2 (10 CHAR)	TAK

Tabela 6.4. Struktura tabeli Payment_method

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
payment_method_key	NUMBER (1)	TAK
payment_method_name	VARCHAR2 (20 CHAR)	TAK

Tabela 6.5 Struktuta tabeli Customer_payment_method

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
customer_key	NUMBER (10)	TAK
payment_method_key	NUMBER (1)	TAK

Tabela 6.6. Struktura tabeli Customer

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
customer_key	NUMBER (10)	TAK
first_name	VARCHAR2 (30 CHAR)	TAK
last_name	VARCHAR2 (50 CHAR)	TAK
account_state	NUMBER (5,2)	TAK
phone_number	NUMBER (12)	TAK
e-mail	VARCHAR2 (50 CHAR)	TAK
PESEL	VARCHAR2 (11 CHAR)	TAK
post_index	VARCHAR2 (6 CHAR)	TAK
city_key	NUMBER (10)	TAK
account_status_key	NUMBER (1)	TAK

Tabela 6.7. Struktura tabeli Bike_station_type

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
bike_station_type_key	NUMBER (1)	TAK
bike_ station_type_name	VARCHAR2 (10 CHAR)	TAK

Tabela 6.8. Struktura tabeli Bike_station

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
bike_station_ key	NUMBER (5)	TAK
bike_ station_name	VARCHAR2 (10 CHAR)	TAK
bike_station_type_key	NUMBER (1)	TAK
city_key	NUMBER (10)	TAK

Tabela 6.9. Struktura tabeli Bike_status

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
bike_status _key	NUMBER (1)	TAK
bike_ status_name	VARCHAR2 (10 CHAR)	TAK

Tabela 6.10. Struktura tabeli Bike_type

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
bike_type _key	NUMBER (1)	TAK
bike_ type_name	VARCHAR2 (10 CHAR)	TAK

Tabela 6.11. Struktura tabeli Bike_subtype

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
bike_subtype _key	NUMBER (2)	TAK
bike_ subtype_name	VARCHAR2 (10 CHAR)	TAK
bike_type_key	NUMBER (1)	TAK

Tabela 6.12. Struktura tabeli Bike

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
bike_key	NUMBER (5)	TAK
bike_ station_key	NUMBER (5)	TAK
bike_status_key	NUMBER (1)	TAK
bike_subtype_key	NUMBER (2)	TAK
bike_code	VARCHAR2 (4 CHAR)	TAK

Tabela 6.13. Struktura tabeli Ride_status

Nazwa kolumny	Typ i rozmiar danych	Wartość wymagana
ride_status _key	NUMBER(1)	TAK
ride_ status_name	VARCHAR2(10 CHAR)	TAK

Tabela 6.14. Struktura tabeli Ride

Nazwa kolumny	Typ i rozmiar danych	Wartosć wymagana
ride_key	NUMBER(10)	TAK
ride_time_start	DATE	TAK
ride_time_finish	DATE	NIE
ride_time	INTEGER	TAK
ride_price	NUMBER(4,2)	TAK
ride_distance	NUMBER(3,1)	TAK
customer_key	NUMBER(10)	TAK
ride_status_key	NUMBER(1)	TAK
bike_station_start_key	NUMBER(5)	TAK
bike_station_finish_key	NUMBER(5)	NIE
bike_key	NUMBER(5)	TAK

6.2. Definicje kluczy głównych, unikatowych i relacyjnych oraz indeksów

Tabela 6.15. Definicje kluczy głównych w poszczególnych tabelach bazy danych

Nazwa tabeli	Nazwa kolumn (y)	Nazwa więzów
Country	country_key	Country_PK
City	city_key	City_PK
Customer	customer_key	Customer_PK
Customer_payment_method	customer_key, payment_method_key	Customer_payment_method_PK
Payment_method	payment_method_key	Payment_method_PK
Bike_station_type	bike_station_type_key	Bike_station_type_PK
Bike_station	bike_station_key	Bike_station_PK
Bike_status	bike_status_key	Bike_status_PK
Bike	bike_key	Bike_PK
Ride	ride_key	Ride_PK
Account_status	account_status_key	Account_status_PK
Ride_status	ride_status_key	Ride_status_PK
Bike_subtype	bike_subtype_key	Bike_subtype_PK
Bike_type	bike_type_key	bike_type_PK

Tabela 6.16. Definicje więzów typu UNIQUE w poszczególnych tabelach bazy danych

Nazwa tabeli	Nazwa kolumny	Nazwa więzów
Country	country_name	Country_name_UN
Country	country_code	Country_code_UN
Customer	phone_number	Customer_phone_UN
Customer	PESEL	Customer_PESEL_UN
Customer	e-mail	Customer_email_UN
Payment_method	payment_method_name	Payment_method_UN
Bike_station_type	bike_station_type_name	Bike_station_type__UN
Bike_status	bike_status_name	Bike_status_UN
Account_status	account_status_name	Account_status_UN
Ride_status	ride_status_name	Ride_status_UN
Bike_subtype	bike_subtype_name	Bike_subtype_UN
Bike_type	bike_type_name	Bike_type_UN

Tabela 6.17. Definicje relacji między poszczególnymi tabelami bazy danych

Tabela pierwotna	Tabela referencyjna	Nazwa kolumny (Foreign Key)	Nazwa więzów
City	Country	country_key	City_Country_FK
Customer	City	city_key	Customer_City_FK
Customer	Account_status	account_status_key	Customer_Account_Status_FK
Customer_payment_method	Customer	customer_key	Customer_FK
Customer_payment_method	Payment_method	payment_method_key	Customer_payment_method_FK
Bike_station	Bike_station_type	bike_station_type_key	Bike_station_type_FK
Bike_station	City	city_key	Bike_station_City_FK
Bike_subtype	Bike_type	bike_type_key	Bike_subtype_Bike_type_FK
Bike	Bike_subtype	bike_subtype_key	Bike_Bike_subtype_FK
Bike	Bike_status	bike_status_key	Bike_Bike_status_FK
Bike	Bike_station	bike_station_key	Bike_Bike_station_FK
Ride	Ride_status	ride_status_key	Ride_status_FK
Ride	Bike	bike_key	Ride_Bike_FK
Ride	Bike_station	bike_station_start_key	Ride_Bike_station_start_FK
Ride	Bike_station	bike_station_finish_key	Ride_Bike_station_finish_FK
Ride	Customer	customer_key	Ride_Customer_FK

Tabela 6.18. Definicje indeksów w poszczególnych tabelach bazy danych

Nazwa tabeli	Nazwa kolumny	Nazwa więzów	Unikatowość rekordów
Bike_station	bike_station_name	Bike_station_name_IDX	NIE
Bike_station	bike_station_type_key	Bike_station_type_IDX	NIE
Ride	ride_time_start	Ride_date_IDX	NIE
Ride	ride_distance	Ride_distance_IDX	NIE

Rozdział 7. KOD SQL – TWORZENIE BAZY DANYCH

```
--tworzenie tabeli "country" z kluczami typu primary key i unique key
      CREATE TABLE country (
            country_key    NUMBER(3)           NOT NULL,
            country_name   VARCHAR2(50 CHAR)   NOT NULL,
            country_code   VARCHAR2(3 CHAR)    NOT NULL
      );
      ALTER TABLE country
            ADD CONSTRAINT country_pk
                  PRIMARY KEY ( country_key );
      ALTER TABLE country
            ADD CONSTRAINT country_code_un
                  UNIQUE ( country_name );
      ALTER TABLE country
            ADD CONSTRAINT country_name_un
                  UNIQUE ( country_name );


--tworzenie tabeli "city" z kluczami typu primary key i unique key
      CREATE TABLE city (
            city_key       NUMBER(10)          NOT NULL,
            city_name      VARCHAR2(50 CHAR)   NOT NULL,
            country_key    NUMBER(3)           NOT NULL
      );
      ALTER TABLE city
            ADD CONSTRAINT city_pk
                  PRIMARY KEY ( city_key );


--tworzenie tabeli "account_status" z kluczami typu primary key i unique key
      CREATE TABLE account_status (
            account_status_key  NUMBER(1)           NOT NULL,
            account_status_name VARCHAR2(10 CHAR)   NOT NULL
      );
      ALTER TABLE account_status
            ADD CONSTRAINT account_status_pk
                  PRIMARY KEY ( account_status_key );
      ALTER TABLE account_status
            ADD CONSTRAINT account_status__un
                  UNIQUE ( account_status_name );
```

```
--tworzenie tabeli "customer" z kluczami typu primary key i unique key
      CREATE TABLE customer (
            customer_key        NUMBER(10)          NOT NULL,
            first_name          VARCHAR2(30 CHAR)   NOT NULL,
            last_name           VARCHAR2(50 CHAR)   NOT NULL,
            account_state       NUMBER(5, 2)        NOT NULL,
            phone_number        NUMBER(12)          NOT NULL,
            "e-mail"            VARCHAR2(50 CHAR)   NOT NULL,
            pesel               VARCHAR2(11 CHAR)   NOT NULL,
            post_index          VARCHAR2(6 CHAR)    NOT NULL,
            city_key            NUMBER(10)          NOT NULL,
            account_status_key  NUMBER(1)           NOT NULL
      );
      ALTER TABLE customer
            ADD CONSTRAINT customer_pk
                  PRIMARY KEY ( customer_key );
      ALTER TABLE customer
            ADD CONSTRAINT customer_email_un
                  UNIQUE ( "e-mail" );
      ALTER TABLE customer
            ADD CONSTRAINT customer_pesel_un
                  UNIQUE ( pesel );
      ALTER TABLE customer
            ADD CONSTRAINT customer_phone_un
                  UNIQUE ( phone_number );


--tworzenie tabeli "payment_method" z kluczami typu primary key i unique key
      CREATE TABLE payment_method (
            payment_method_key  NUMBER(1)           NOT NULL,
            payment_method_name VARCHAR2(20 CHAR)   NOT NULL
      );
      ALTER TABLE payment_method
            ADD CONSTRAINT payment_method_pk
                  PRIMARY KEY ( payment_method_key );
      ALTER TABLE payment_method
            ADD CONSTRAINT payment_method_un
                  UNIQUE ( payment_method_name );


--tworzenie tabeli "customer_payment_method" z kluczami primary i unique key
      CREATE TABLE customer_payment_method (
            customer_key        NUMBER(10)     NOT NULL,
            payment_method_key  NUMBER(1)      NOT NULL
      );
      ALTER TABLE customer_payment_method
            ADD CONSTRAINT customer_payment_method_pk
                  PRIMARY KEY ( customer_key, payment_method_key );
```

```
--tworzenie tabeli "bike_station_type" z kluczami typu primary i unique key
      CREATE TABLE bike_station_type (
           bike_station_type_key    NUMBER(1)           NOT NULL,
           bike_station_type_name   VARCHAR2(10 CHAR)   NOT NULL
      );
      ALTER TABLE bike_station_type
           ADD CONSTRAINT bike_station_type_pk
                PRIMARY KEY ( bike_station_type_key );
      ALTER TABLE bike_station_type
           ADD CONSTRAINT bike_station_type__un
                UNIQUE ( bike_station_type_name );


--tworzenie tabeli "bike_station" z kluczami typu primary key i unique key
      CREATE TABLE bike_station (
           bike_station_key         NUMBER(5)           NOT NULL,
           bike_station_name        VARCHAR2(10 CHAR)   NOT NULL,
           bike_station_type_key    NUMBER(1)           NOT NULL,
           city_key                 NUMBER(10)          NOT NULL
      );
      ALTER TABLE bike_station
           ADD CONSTRAINT bike_station_pk
                PRIMARY KEY ( bike_station_key );


--tworzenie tabeli "bike_type" z kluczami typu primary key i unique key
      CREATE TABLE bike_type (
           bike_type_key  NUMBER(1)           NOT NULL,
           bike_type_name VARCHAR2(10 CHAR)   NOT NULL
      );
      ALTER TABLE bike_type
           ADD CONSTRAINT bike_type_pk
                PRIMARY KEY ( bike_type_key );
      ALTER TABLE bike_type
           ADD CONSTRAINT bike_type__un
                UNIQUE ( bike_type_name );
```

```
--tworzenie tabeli "bike_subtype" z kluczami typu primary key i unique key
      CREATE TABLE bike_subtype (
            bike_subtype_key    NUMBER(2)           NOT NULL,
            bike_subtype_name   VARCHAR2(10 CHAR)   NOT NULL,
            bike_type_key       NUMBER(1)           NOT NULL
      );
      ALTER TABLE bike_subtype
            ADD CONSTRAINT bike_subtype_pk
                  PRIMARY KEY ( bike_subtype_key );
      ALTER TABLE bike_subtype
            ADD CONSTRAINT bike_subtype__un
                  UNIQUE ( bike_subtype_name );
--tworzenie tabeli "bike_status" z kluczami typu primary key i unique key
      CREATE TABLE bike_status (
            bike_status_key     NUMBER(1)           NOT NULL,
            bike_status_name    VARCHAR2(10 CHAR)   NOT NULL
      );
      ALTER TABLE bike_status
            ADD CONSTRAINT bike_status_pk
                  PRIMARY KEY ( bike_status_key );
      ALTER TABLE bike_status
            ADD CONSTRAINT bike_status__un
                  UNIQUE ( bike_status_name );


--tworzenie tabeli "bike" z kluczami typu primary key i unique key
      CREATE TABLE bike (
            bike_key            NUMBER(5) NOT NULL,
            bike_station_key    NUMBER(5) NOT NULL,
            bike_status_key     NUMBER(1) NOT NULL,
            bike_subtype_key    NUMBER(2) NOT NULL
      );
      ALTER TABLE bike
            ADD CONSTRAINT bike_pk
                  PRIMARY KEY ( bike_key );


--tworzenie tabeli "ride_status" z kluczami typu primary key i unique key
      CREATE TABLE ride_status (
            ride_status_key     NUMBER(1)           NOT NULL,
            ride_status_name    VARCHAR2(10 CHAR)   NOT NULL
      );
      ALTER TABLE ride_status
            ADD CONSTRAINT ride_status_pk
                  PRIMARY KEY ( ride_status_key );
      ALTER TABLE ride_status
            ADD CONSTRAINT ride_status_un
                  UNIQUE ( ride_status_name );
```

```
--tworzenie tabeli "ride" z kluczami typu primary key i unique key
      CREATE TABLE ride (
            ride_key                NUMBER(10)     NOT NULL,
            ride_time_start         DATE           NOT NULL,
            ride_time_finish        DATE,
            ride_time               INTEGER        NOT NULL,
            ride_price              NUMBER(2, 4)   NOT NULL,
            customer_key            NUMBER(10)     NOT NULL,
            ride_status_key         NUMBER(1)      NOT NULL,
            bike_station_start_key  NUMBER(5)      NOT NULL,
            bike_station_finish_key NUMBER(5),
            bike_key                NUMBER(5)      NOT NULL
      );
      ALTER TABLE ride
            ADD CONSTRAINT ride_pk
                  PRIMARY KEY ( ride_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "bike" a "bike_station"
      ALTER TABLE bike
            ADD CONSTRAINT bike_bike_station_fk
                  FOREIGN KEY ( bike_station_key )
                  REFERENCES bike_station ( bike_station_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "bike" a "bike_status"
      ALTER TABLE bike
            ADD CONSTRAINT bike_bike_status_fk
                  FOREIGN KEY ( bike_status_key )
                        REFERENCES bike_status ( bike_status_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "bike" a "bike_subtype"
      ALTER TABLE bike
            ADD CONSTRAINT bike_bike_subtype_fk
                  FOREIGN KEY ( bike_subtype_key )
                           REFERENCES bike_subtype ( bike_subtype_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "bike_station" a "city"
      ALTER TABLE bike_station
            ADD CONSTRAINT bike_station_city_fk
                  FOREIGN KEY ( city_key )
                        REFERENCES city ( city_key );
```

```
--tworzenie klucza typu foreign key pomiędzy tabelami "bike_station" a
--bike_station_type"
      ALTER TABLE bike_station
            ADD CONSTRAINT bike_station_type_fk
                  FOREIGN KEY ( bike_station_type_key )
                        REFERENCES bike_station_type ( bike_station_type_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "bike_subtype" a
--"bike_type"
      ALTER TABLE bike_subtype
            ADD CONSTRAINT bike_subtype_bike_type_fk
                  FOREIGN KEY ( bike_type_key )
                        REFERENCES bike_type ( bike_type_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "city" a "country"
      ALTER TABLE city
            ADD CONSTRAINT city_country_fk
                  FOREIGN KEY ( country_key )
                        REFERENCES country ( country_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "customer" a
--"account_status"
      ALTER TABLE customer
            ADD CONSTRAINT customer_account_status_fk
                  FOREIGN KEY ( account_status_key )
                        REFERENCES account_status ( account_status_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "customer" a "city"
      ALTER TABLE customer
            ADD CONSTRAINT customer_city_fk
                  FOREIGN KEY ( city_key )
                        REFERENCES city ( city_key );


--tworzenie klucza typu foreign key pomiędzy tabelami
--"customer_payment_method" a "customer"
      ALTER TABLE customer_payment_method
            ADD CONSTRAINT customer_fk
                  FOREIGN KEY ( customer_key )
                        REFERENCES customer ( customer_key );
```

```
--tworzenie klucza typu foreign key pomiędzy tabelami
--"customer_payment_method" a "payment_method"
      ALTER TABLE customer_payment_method
            ADD CONSTRAINT customer_payment_method_fk
                  FOREIGN KEY ( payment_method_key )
                        REFERENCES payment_method ( payment_method_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "ride" a "bike"
      ALTER TABLE ride
            ADD CONSTRAINT ride_bike_fk
                  FOREIGN KEY ( bike_key )
                        REFERENCES bike ( bike_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "ride" a "bike_station"
      ALTER TABLE ride
            ADD CONSTRAINT ride_bike_station_finish_fk
                  FOREIGN KEY ( bike_station_finish_key )
                        REFERENCES bike_station ( bike_station_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "ride" a "bike_station"
      ALTER TABLE ride
            ADD CONSTRAINT ride_bike_station_start_fk
                  FOREIGN KEY ( bike_station_start_key )
                        REFERENCES bike_station ( bike_station_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "ride" a "customer"
      ALTER TABLE ride
            ADD CONSTRAINT ride_customer_fk
                  FOREIGN KEY ( customer_key )
                        REFERENCES customer ( customer_key );


--tworzenie klucza typu foreign key pomiędzy tabelami "ride" a "ride_status"
      ALTER TABLE ride
            ADD CONSTRAINT ride_status_fk
                  FOREIGN KEY ( ride_status_key )
                        REFERENCES ride_status ( ride_status_key );
```

```
--tworzenie indeksów w tabeli "Bike_station"
      CREATE INDEX bike_station_name_idx ON
           bike_station (
                bike_station_name
           ASC );
      CREATE INDEX bike_station_type_idx ON
           bike_station (
                bike_station_type_key
           ASC );

--tworzenie indeksów w tabeli "Ride"
      CREATE INDEX ride_date_idx ON
           ride (
                ride_time_start
           ASC );
      CREATE INDEX ride_distance_idx ON
           ride (
                ride_distance
           ASC );
```

Raport powstał podczas zajęć laboratoryjnych z przedmiotu prowadzonego w ramach projektu *„Zintegrowany Program Rozwoju Politechniki Lubelskiej – część druga”*, umowa nr **POWR.03.05.00-00-Z060/18-00** w ramach Programu Operacyjnego Wiedza Edukacja Rozwój 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Społecznego

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Unia Europejska Europejski Fundusz Społeczny