# 

Rower Miejski

Baza danych aplikacji będzie zawierała:

* Użytkowników
* Ceny
* Rowery
* Stacje
* Wypożyczenia

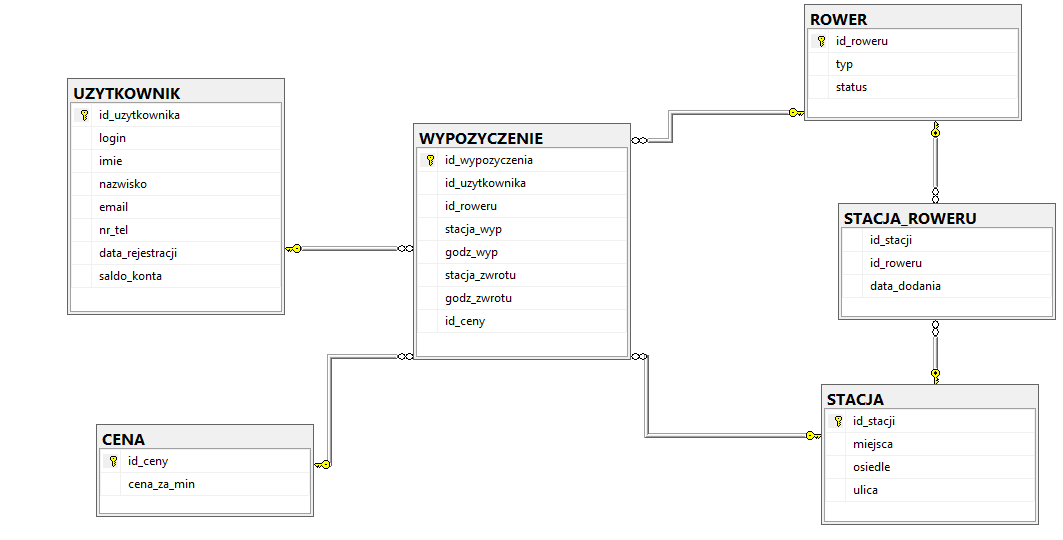
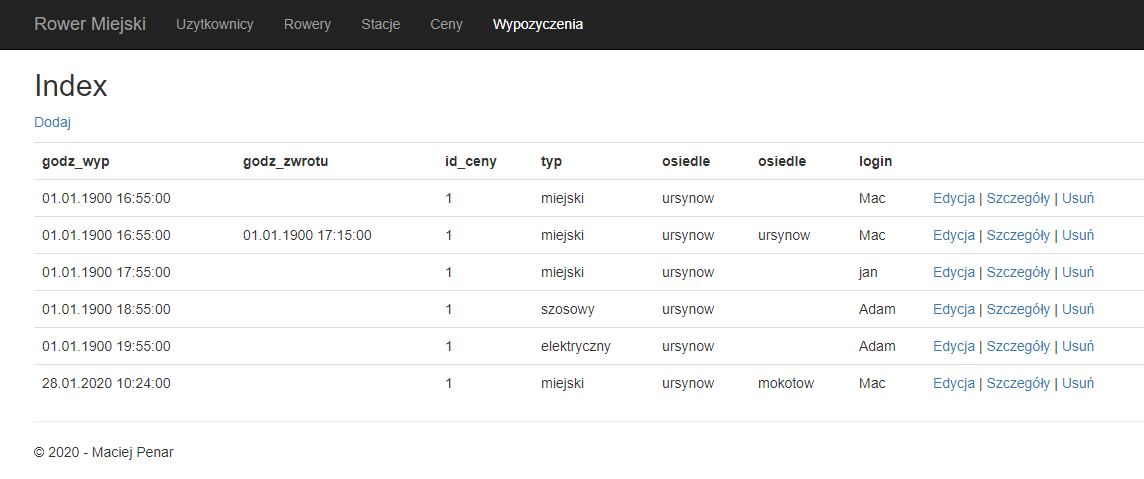


Diagram ERD

Aplikacja posiada możliwość dodawania, usuwania, edytowania tabel oraz wyświetlania szczegółowych danych.

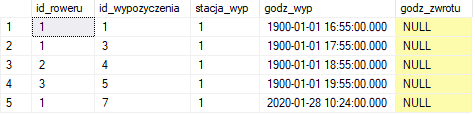


Wygląd tabeli WYPOŻYCZENIA

Zapytania:

1. **Skąd wiadomo, że rower jest wypożyczony lub wolny**

SELECT r.id\_roweru, w.id\_wypozyczenia, w.stacja\_wyp, w.godz\_wyp, w.godz\_zwrotu FROM Wypozyczenie w JOIN ROWER r ON r.id\_roweru=w.id\_roweru WHERE godz\_zwrotu is null



1. **Ile rowerów jest dostępnych na stacji nr 1**

SELECT sr.id\_roweru, r.status From STACJA\_ROWERU sr JOIN ROWER r ON sr.id\_roweru=r.id\_roweru WHERE sr.id\_stacji = 1 AND r.status='wolny'

C:\Users\Maciej\AppData\Local\Microsoft\Windows\INetCache\Content.Word\wynik zapytania nr 2.png

1. **Imiona użytkowników którzy wypożyczyli rower typu 'miejski'**

Select u.imie, r.typ From UZYTKOWNIK u JOIN WYPOZYCZENIE w ON u.id\_uzytkownika=w.id\_uzytkownika JOIN ROWER r ON r.id\_roweru=w.id\_roweru Where r.typ='miejski'



**Skrypt tworzący bazę danych:**

CREATE TABLE UZYTKOWNIK

(

id\_uzytkownika int PRIMARY KEY,

login varchar(25) UNIQUE NOT NULL,

imie varchar(25) not null,

nazwisko varchar(25) not null,

email varchar(40) unique not null,

nr\_tel int not null,

data\_rejestracji datetime not null,

saldo\_konta money not null default 0

);

Create table CENA

(

id\_ceny int primary key,

cena\_za\_min money not null

);

Create table ROWER

(

id\_roweru int primary key,

typ varchar(25) not null,

status varchar(25) default 'wolny'

CONSTRAINT ck\_status\_roweru CHECK (status in('wolny','zajety'))

);

Create table STACJA

(

id\_stacji int primary key,

miejsca int not null,

osiedle varchar(40),

ulica varchar(40)

);

Create table WYPOZYCZENIE

(

id\_wypozyczenia int primary key,

id\_uzytkownika int REFERENCES UZYTKOWNIK(id\_uzytkownika),

id\_roweru int REFERENCES ROWER(id\_roweru),

stacja\_wyp int REFERENCES STACJA(id\_stacji),

godz\_wyp datetime not null,

stacja\_zwrotu int REFERENCES STACJA(id\_stacji),

godz\_zwrotu datetime,

id\_ceny int REFERENCES CENA(id\_ceny),

);

Create table STACJA\_ROWERU

(

id\_stacji int REFERENCES STACJA(id\_stacji),

id\_roweru int REFERENCES ROWER(id\_roweru),

data\_dodania datetime

);

**Dodanie przykładowych danych:**

INSERT INTO UZYTKOWNIK VALUES (1, 'Mac', 'Maciej', 'Penar', 'mac@wp.pl', 123456, '05/12/2019', '');

INSERT INTO UZYTKOWNIK VALUES (2, 'jan', 'Jan', 'Nowak', 'nowak@wp.pl', 123456789, '06/12/2019', '');

INSERT INTO UZYTKOWNIK VALUES (3, 'Adam', 'Adam', 'Malysz', 'adam@wp.pl', 453256, '05/12/2019', '');

INSERT INTO ROWER VALUES (1,'miejski','wolny');

INSERT INTO ROWER VALUES (2,'szosowy','wolny');

INSERT INTO ROWER VALUES (3,'elektryczny','zajety');

INSERT INTO STACJA VALUES (1,30,'ursynow','dluga 5');

INSERT INTO STACJA VALUES (2,50,'mokotow','zielona 11');

INSERT INTO STACJA\_ROWERU VALUES (1,2,'04/12/2019')

INSERT INTO STACJA\_ROWERU VALUES (1,3,'06/12/2019')

INSERT INTO STACJA\_ROWERU VALUES (2,1,'05/12/2019')

INSERT INTO CENA VALUES (1, 5)

INSERT INTO CENA VALUES (2, 2)

INSERT INTO WYPOZYCZENIE (id\_wypozyczenia, id\_uzytkownika, id\_roweru, stacja\_wyp, godz\_wyp, id\_ceny)

VALUES (1,1,1,1,'16:55',1)

INSERT INTO WYPOZYCZENIE (id\_wypozyczenia, id\_uzytkownika, id\_roweru, stacja\_wyp, godz\_wyp, stacja\_zwrotu, godz\_zwrotu, id\_ceny)

VALUES (2,1,1,1,'16:55',1,'17:15',1)

INSERT INTO WYPOZYCZENIE (id\_wypozyczenia, id\_uzytkownika, id\_roweru, stacja\_wyp, godz\_wyp, id\_ceny)

VALUES (3,2,1,1,'17:55',1)

INSERT INTO WYPOZYCZENIE (id\_wypozyczenia, id\_uzytkownika, id\_roweru, stacja\_wyp, godz\_wyp, id\_ceny)

VALUES (4,3,2,1,'18:55',1)

INSERT INTO WYPOZYCZENIE (id\_wypozyczenia, id\_uzytkownika, id\_roweru, stacja\_wyp, godz\_wyp, id\_ceny) VALUES (5,3,3,1,'19:55',1)