

Wydział Elektroniki, Automatyki i Informatyki Politechniki Świętokrzyskiej		
Bazy danych 1 - MiniProjekt		
Data wysłania projektu: 30.01.2022 Grupa 2ID12A	Temat: Pogotowie ratownicze	Członkowie zespołu: <ul style="list-style-type: none"> • Arkadiusz Wolski • Sebastian Wacek • Jakub Sokół 13B

Tematem naszego projektu było pogotowie ratownicze, którego bazę danych należało stworzyć.

Technologie których używaliśmy przy tworzeniu naszego projektu:

- Oracle SQL Developer/Oracle LIVE SQL – do tworzenia oraz testowania bazy danych.
- Baza danych Oracle.
- app.diagrams.net – do tworzenia diagramu przedstawiającego związki encji w naszym projekcie.
- Notepad++ - do stworzenia strony internetowej w języku HTML oraz skryptów odpowiedzialnych za połączenie z bazą danych w języku PHP.
- xampp – do obsługi języka PHP i HTML.
- Discord – do komunikacji podczas prac nad projektem.
- GitHub – do zobrazowania efektów prac nad projektem oraz wkładu poszczególnych członków grupy.

W naszej bazie danych zastosowaliśmy elementy takie jak:

- 12 tabel z różnego rodzaju relacjami.
- 6 widoków pozwalających na przeglądanie bazy danych z różnych perspektyw.
- 4 kursory użyte w różnych procedurach.
- 3 wyzwalacze odpowiadające za operacje na bazie danych pod wpływem różnych zdarzeń.

Podział obowiązków przy pracy nad projektem:

Aby zoptymalizować czas i efektywność prac nad projektem, zastosowaliśmy następujący podział obowiązków:

Arkadiusz Wolski:

- Skrypty PHP,
- Kursory,
- Instrukcje INSERT

Sebastian Wacek:

- Strona internetowa,
- Wyzwalacze,
- Sprawozdanie

Jakub Sokół:

- Diagram związku encji
- Widoki
- Instrukcje CREATE

Każdy z członków zespołu angażował się w projekt oraz oprócz przydzielonych obowiązków pracował nad każdym z elementów projektu.

Diagram związków encji:

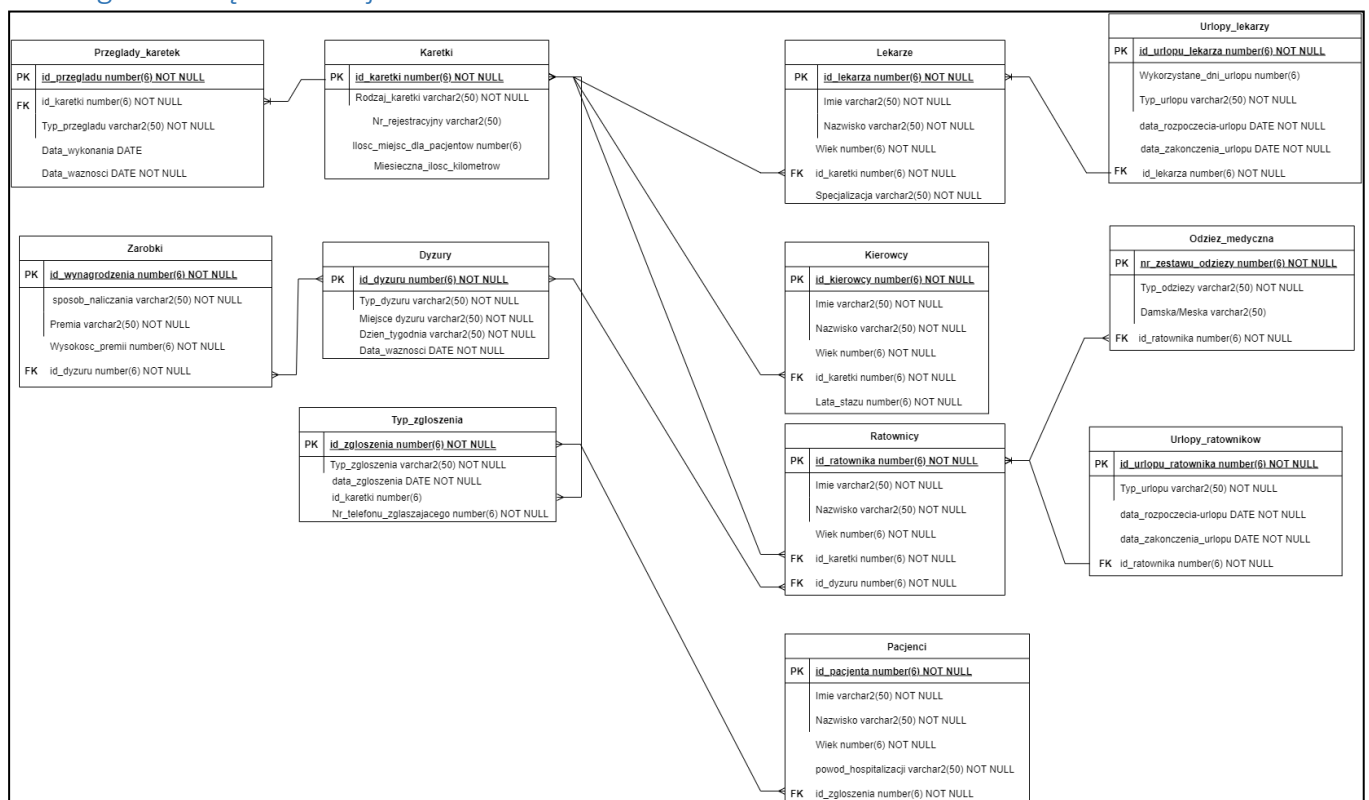


Diagram przedstawia 12 tabel które zawiera nasz projekt oraz 11 relacji różnego rodzaju.

Widoki:

```
-- 1.

CREATE OR REPLACE VIEW pacjenci_zgloszenia AS
SELECT pa.imie, pa.nazwisko, pa.wiek, pa.powod_hospitalizacji, tz.data_zgloszenia, tz.nr_telefonu_zglaszajacego
FROM pacjenci pa, typ_zgloszenia tz
WHERE pa.id_zgloszenia = tz.id_zgloszenia;
```

Pierwszy widok wyświetla imiona, nazwiska, wiek i powód hospitalizacji w zestawieniu z datą zgłoszenia i nr telefonu gdzie id pacjenta jest równe z id zgłoszenia.

```
-- 2.

CREATE OR REPLACE VIEW karetki_view AS
SELECT rodzaj_karetki, nr_rejestracyjny, ilosc_miejsc_dla_pacjentow, miesieczna_liczba_kilometrow
FROM karetki;
```

Drugi widok pokazuje rodzaje karetek, ich numery rejestracyjne, ilość miejsc dla pacjentów i liczbę kilometrów którą przejechały w ostatnim miesiącu.

```
-- 3.

CREATE OR REPLACE VIEW dyzury_view AS
SELECT typ_dyzyru, miejsce_dyzyru, dzien_tygodnia, data_waznosci
FROM dyzury;
```

Trzeci z widoków pokazuje informacje o dyżurach, takie jak typ, miejsce, dzień tygodnia oraz data dyżuru.

```
-- 4.

CREATE OR REPLACE VIEW kierowcy_karetki AS
SELECT kier.imie, kier.nazwisko, kier.wiek, kier.lata_stazu, kar.rodzaj_karetki, kar.nr_rejestracyjny, kar.ilosc_miejsc_dla_pacjentow, kar.miesieczna_liczba_kilometrow
FROM karetki kar, kierowcy kier WHERE kar.id_karetki = kier.id_karetki;
```

Kolejny – czwarty widok, przedstawia imię, nazwisko, wiek oraz lata stażu kierowców oraz informacje o karetkach do których są przypisani.

```
-- 5.

CREATE OR REPLACE VIEW zarobki_view AS
SELECT wysokosc_wynagrodzenia, wysokosc_premii, sposob_naliczania
FROM zarobki;
```

Piąty widok, przedstawia wysokość wynagrodzenia, wysokość premii i sposób naliczania premii z tabeli zarobki

```
-- 6.

CREATE OR REPLACE VIEW ratownicy_odziez AS
SELECT ra.imie, ra.nazwisko, om.Typ_odziezy, om.Damska_czy_meska
FROM Ratownicy ra, Odziez_medyczna om
WHERE ra.id_ratownika = om.nr_zestawu_odziezy;
```

Ostatni widok obrazuje odzież ratowników, wyświetla imię i nazwisko ratownika oraz dane o przypisanej mu odzieży.

Kursory:

```
-- 1.

CREATE OR REPLACE PROCEDURE zwiekszenie_wynagrodzenia_na_podst_wieku(min_wymagany_wiek NUMBER DEFAULT 0, kwota_podwyzszenia NUMBER DEFAULT 0)
AS
    CURSOR ratownik_zarobki_cursor IS
        SELECT dy.id_dyzuru, rt.imie, rt.nazwisko, rt.wiek, dy.dzien_tygodnia, zr.wysokosc_wynagrodzenia, zr.wysokosc_premii
        FROM ratownicy rt, dyzury dy, zarobki zr
        WHERE rt.id_dyzuru = dy.id_dyzuru AND dy.id_dyzuru = zr.id_dyzuru;

BEGIN
    FOR tmp IN ratownik_zarobki_cursor LOOP
        IF tmp.wiek >= min_wymagany_wiek THEN
            UPDATE zarobki z
            SET z.wysokosc_premii = z.wysokosc_premii + kwota_podwyzszenia
            WHERE tmp.id_dyzuru = z.id_dyzuru;
        END IF;
    END LOOP;
END;
/

SELECT * FROM ratownik_zarobki;
EXEC zwiekszenie_wynagrodzenia_na_podst_wieku(27, 20);
SELECT * FROM ratownik_zarobki;
```

Pierwszy z kursorów został użyty w procedurze, która podwyższa wynagrodzenie ratowników na podstawie ich wieku o kwotę podwyższenia, obie te wartości należy przekazać w parametrze wywołania procedury.

```
-- 2.

CREATE OR REPLACE PROCEDURE zwieksz_wynagrodzenie_o_premie
AS
    CURSOR wszystkie_zarobki IS
        SELECT * FROM zarobki
        FOR UPDATE;

BEGIN
    FOR tmp IN wszystkie_zarobki LOOP
        UPDATE zarobki
        SET wysokosc_wynagrodzenia = wysokosc_wynagrodzenia + wysokosc_premii
        WHERE CURRENT OF wszystkie_zarobki;
    END LOOP;
END;
/

SELECT * FROM zarobki;
EXEC zwieksz_wynagrodzenie_o_premie;
SELECT * FROM zarobki;
```

Drugi kursor został wykorzystany w procedurze, która zwiększa wynagrodzenie o premię. Kursor pobiera wszystkie dane z tabeli zarobki a następnie jest użyty w pętli for, która oblicza nową wysokość wynagrodzenia.

```
-- 3.

CREATE OR REPLACE PROCEDURE zwiekszenie_premii_weekend(kwota_podwyzszenia NUMBER DEFAULT 0)
AS
    CURSOR ratownik_zarobki_cursor IS
    SELECT dy.id_dyzuru, rt.imie, rt.nazwisko, rt.wiek, dy.dzien_tygodnia, zr.wysokosc_wynagrodzenia, zr.wysokosc_premii
    FROM ratownicy rt, dyzury dy, zarobki zr
    WHERE rt.id_dyzuru = dy.id_dyzuru AND dy.id_dyzuru = zr.id_dyzuru;

BEGIN
    FOR tmp IN ratownik_zarobki_cursor LOOP
        IF tmp.dzien_tygodnia = 'sobota' OR tmp.dzien_tygodnia = 'niedziela' THEN
            UPDATE zarobki z
            SET z.wysokosc_premii = z.wysokosc_premii + kwota_podwyzszenia
            WHERE tmp.id_dyzuru = z.id_dyzuru;
        END IF;
    END LOOP;

END;
/

SELECT * FROM ratownik_zarobki;
EXEC zwiekszenie_premii_weekend(100);
SELECT * FROM ratownik_zarobki;
```

Trzeci kursor wykorzystany został wewnątrz procedury, która na podstawie dnia tygodnia wylicza dodatkowe wynagrodzenie, jeśli jest to weekend. Kursor pobiera zarobki ratowników, na podstawie ich dyżurów a następnie procedura w pętli for oblicza dodatkowe wynagrodzenie. Jeśli dzień tygodnia to sobota lub niedziela, podwyższa wynagrodzenie o kwotę podaną w parametrze wywołania procedury.

```
-- 4.

CREATE OR REPLACE PROCEDURE przeglad_karetek_po_terminie
AS
    CURSOR przeglad_cursor IS
    SELECT *
    FROM przeglad_karetek;

    current_date DATE;

BEGIN
    SELECT SYSDATE
    INTO current_date
    FROM DUAL;

    DBMS_OUTPUT.PUT_LINE('Karetki, ktorym nalezy zrobic przeglad:');
    DBMS_OUTPUT.PUT_LINE('ID karetki' || ' | ' | ' || ' Kiedy przeglad sie skonczyl');
    FOR tmp IN przeglad_cursor LOOP
        IF tmp.data_waznosci < current_date THEN
            DBMS_OUTPUT.PUT_LINE( tmp.id_karetki || ' | ' | ' || tmp.data_waznosci);
        END IF;
    END LOOP;

END;
/

EXEC przeglad_karetek_po_terminie ;
```

Ostatni z kursorów został użyty w procedurze, która sprawdza terminy ważności przeglądów karetek, jeśli termin jest mniejszy od aktualnej daty, wyświetla odpowiedni komunikat. Kursor pobiera wszystkie dane z tabeli przeglad_karetek.

Wyzwalacze:

```
-- 1.
CREATE SEQUENCE karetka_seq
START WITH 15;

CREATE OR REPLACE TRIGGER karetka_trigger
INSTEAD OF INSERT ON karetka_view
FOR EACH ROW
BEGIN
INSERT INTO karetka VALUES (karetka_seq.nextval, :NEW.rodzaj_karetka, :NEW.nr_rejestacyjny, :NEW.ilosc_miejsc_dla_pacjentow, :NEW.miesieczna_liczba_kilometrow);
END;
/
```

Pierwszy z wyzwalaczy odpowiada za obsługę widoku karetka_view. Trigger zostaje wykonany zamiast instrukcji INSERT, kiedy ta, zostanie wprowadzona przez użytkownika. Działa tak samo jak instrukcja INSERT, a klucz główny pobiera z sekwencji.

```
-- 2.
CREATE SEQUENCE zgloszenia_seq
START WITH 15;

CREATE OR REPLACE TRIGGER zgloszenia_trigger
INSTEAD OF INSERT ON typ_zgloszenia_view
FOR EACH ROW
BEGIN
INSERT INTO typ_zgloszenia VALUES (zgloszenia_seq.nextval, :NEW.typ_zgloszenia, :NEW.data_zgloszenia, :NEW.nr_telefonu_zglaszajacego, :NEW.id_karetka);
END;
/
```

Drugi wyzwalacz, wykonuje się wtedy, kiedy użytkownik będzie chciał dodać wiersz do tabeli zgłoszenia. Operuje na widoku typ_zgloszenia_view, wartość klucza głównego pobiera z sekwencji.

Ostatni wyzwalacz operuje na widoku dyzury_view, wykonuje się zamiast instrukcji INSERT i działa w taki sam sposób jak ona, wartość klucza głównego pobiera z sekwencji dyzury_seq.

```
-- 3.
CREATE SEQUENCE dyzury_seq
START WITH 15;

CREATE OR REPLACE TRIGGER dyzury_trigger
INSTEAD OF INSERT ON dyzury_view
FOR EACH ROW
BEGIN
INSERT INTO dyzury VALUES (dyzury_seq.nextval, :NEW.typ_dyzyru, :NEW.miejsce_dyzyru, :NEW.dzien_tygodnia, :NEW.data_waznosci);
END;
/
```

Lista funkcjonalności:

- Obsługa danych bazy pogotowia ratowniczego
- Możliwość naliczenia dodatkowego wynagrodzenia na podstawie wieku oraz przepracowanych dni.
- Możliwość sprawdzenia czy dana karetka ma ważny przegląd, lub kiedy się kończy jego data ważności.
- Różne widoki bazy danych pozwalają na jej obsługę z różnych stanowisk administracyjnych placówki.
- Dodawanie rekordów do tabel za pomocą strony internetowej.

Działanie aplikacji:

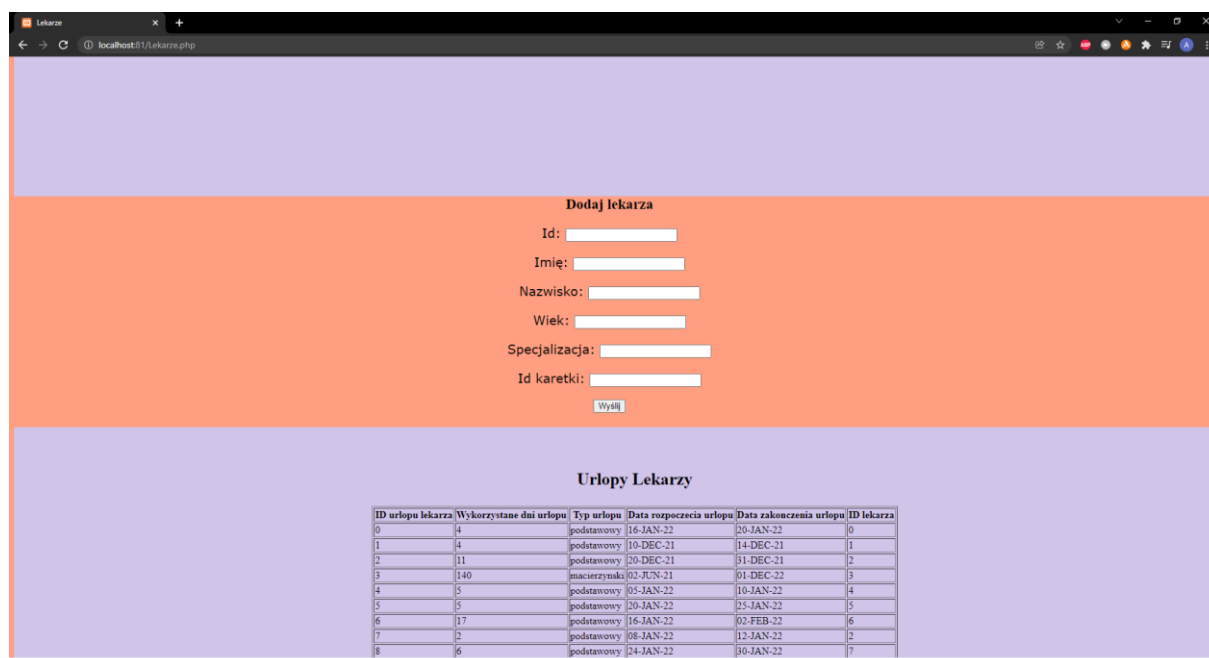
Nasza strona internetowa posiada zakładki, które przekierowują na konkretne podstrony, na których wyświetlane są odpowiednie tabele. Tabele są znormalizowane oraz wyświetlane w centralnej części strony. Istnieje również możliwość dodawania nowych rekordów do tabel. Kilka przykładów:



The screenshot shows a web application titled "Pogotowie Ratunkowe" with a navigation bar containing links: Start, Lekarze, Ratownicy, Kierowcy, Pacjenci, Karetki, Dyżury, Odsiecz medyczna, and Zarobki. Below the navigation bar is a "Zgłoszenia" button. The main content area displays a table titled "Lekarze" with the following data:

ID lekarza	Imię	Nazwisko	Wiek	Specjalizacja	ID karetki
0	Czesław	Kurek	57	Kardiolog	0
1	Albert	Szymański	46	Pediatra	0
2	Olivier	Orzechowski	38	Pediatra	1
3	Lucyna	Jóźwiak	30	Ortopeda	2
4	Patrycja	Mroz	29	Pediatra	2
5	Oskar	Wysocki	34	Kardiolog	3
6	Antoni	Sokołowski	36	Ortopeda	4
7	Herbert	Konieczny	39	Ortopeda	4
8	Dominika	Owczarek	39	Pediatra	4
9	Hanna	Kwiatkowska	44	Kardiolog	7
10	Klaudiusz	Karpinski	43	Ortopeda	7
11	Alfred	Czerwinski	35	Kardiolog	9
12	Nadia	Andrzejewska	45	Pediatra	11
13	Mateusz	Kruk	38	Kardiolog	13
14	Liliana	Kowalewska	41	Pediatra	5
15	Adam	Kowalski	22	Endokrynolog	7
16	Karol	Nitkiewicz	29	Pediatra	9
17	Aa	bb	22	hh	1

Tabela lekarze



The screenshot shows the "Dodaj lekarza" form with input fields for Id, Imię, Nazwisko, Wiek, Specjalizacja, and Id karetki, followed by a "Wyślij" button. Below the form is a table titled "Urlopy Lekarzy" with the following data:

ID urlopu lekarza	Wykorzystane dni urlopu	Typ urlopu	Data rozpoczęcia urlopu	Data zakończenia urlopu	ID lekarza
0	4	podstawowy	16-JAN-22	20-JAN-22	0
1	4	podstawowy	10-DEC-21	14-DEC-21	1
2	11	podstawowy	20-DEC-21	31-DEC-21	2
3	140	macierzyński	02-JUN-21	01-DEC-22	3
4	5	podstawowy	05-JAN-22	10-JAN-22	4
5	5	podstawowy	20-JAN-22	25-JAN-22	5
6	17	podstawowy	16-JAN-22	02-FEB-22	6
7	2	podstawowy	08-JAN-22	12-JAN-22	2
8	6	podstawowy	24-JAN-22	30-JAN-22	7

Formularz pozwalający na dodanie lekarza do bazy

Kierowcy

localhost:81/kierowcy.php

Pogotowie Ratunkowe

Start Lekarze Ratownicy Kierowcy Pacjenci Karetki Dyżury Oddział medyczna Zarobki

Zgłoszenia

Kierowcy

ID kierowcy	Imie	Nazwisko	Wiek	Lata stażu	ID karetki
0	Radosław	Głowacki	34	6	0
1	Marcel	Dobrowolski	37	6	1
2	Maciej	Andrzejewski	28	8	2
3	Aleksander	Tomaszewski	29	4	3
4	Juliusz	Mazurkiewicz	32	7	4
5	Alfred	Wrona	33	9	5
6	Zbigniew	Stankiewicz	41	13	6
7	Malgorzata	Stankiewicz	37	10	7
8	Wiktoria	Kozioł	40	13	8
9	Halina	Stasiak	43	12	9
10	Aleksandra	Witkowska	33	4	10
11	Lucja	Jasnińska	29	2	11
12	Leonard	Czajkowski	28	1	12
13	Juliusz	Zak	32	3	13
14	Hubert	Bielecki	37	5	14

Tabela kierowcy

Kierowcy

localhost:81/kierowcy.php

3	Aleksander	Tomaszewski	29	4	3
4	Juliusz	Mazurkiewicz	32	7	4
5	Alfred	Wrona	33	9	5
6	Zbigniew	Stankiewicz	41	13	6
7	Malgorzata	Stankiewicz	37	10	7
8	Wiktoria	Kozioł	40	13	8
9	Halina	Stasiak	43	12	9
10	Aleksandra	Witkowska	33	4	10
11	Lucja	Jasnińska	29	2	11
12	Leonard	Czajkowski	28	1	12
13	Juliusz	Zak	32	3	13
14	Hubert	Bielecki	37	5	14

Dodaj kierowcę

Id:

Imie:

Nazwisko:

Wiek:

Lata stażu:

ID karetki:

Formularz pozwalający dodać nowego kierowcę do bazy

Testy jednostkowe:

Test działania widoku pacjenci_zgloszenia:

1	SELECT * FROM pacjenci_zgloszenia;					
IMIE	NAZWISKO	WIEK	POWOD_HOSPITALIZACJI	DATA_ZGLOSZENIA	NR_TELEFONU_ZGLASZAJACEGO	
Andrzej	Gala	64	Uraz nogi	12-JAN-22	+48 103948573	
Mieczysław	Kowalska	54	Uraz ręki	12-JAN-22	847593759	
Gracjan	Orłowski	78	Uraz nogi	17-JAN-22	495739502	
Władysław	Jakubowski	24	Omdlenie	15-DEC-21	+48 103856305	
Ksawery	Lipiński	36	Choroba serca	02-JAN-22	+48 947103548	
Anna	Milewska	48	Omdlenie	07-JAN-22	037503456	
Sylwester	Górecki	65	Choroba serca	10-JAN-22	103994758	
Klaudiusz	Jabłoński	78	Omdlenie	23-DEC-21	937405883	
Mateusz	Wiśniewski	81	Zapaść	29-JAN-22	102875884	
Michalina	Laskowska	83	Omdlenie	21-JAN-22	102956553	
Patryk	Matuszewski	37	Omdlenie	07-JAN-22	576990746	

Test działania Procedury pokazującej karetki które wymagają przeglądu:

1	EXEC przeglad_karetek_po_terminie;
2	

Statement processed.

Karetki, ktorým należy zrobic przeglad:

ID karetki	Kiedy przeglad sie skonczyl
9	24-JAN-22
12	20-JAN-22

Test działania wyzwalacza dodającego karetki do widoku karetki_view po dodaniu wiersza do tabeli Karetki:

Efekt działania polecenia `SELECT * FROM karetki_view;` przed dodaniem nowego wiersza do tabeli:

RODZAJ_KARETKI	NR_REJESTRACYJNY	ILOSC_MIEJSC_DLA_PACJENTOW	MIESIECZNA_LICZBA_KILOMETROW
podstawowa	BHA8527	1	1053
podstawowa	NGI5403	1	976
specjalistyczna	WS29804	1	983
specjalistyczna	NBR7134	1	983
specjalistyczna	CBY7449	1	1167
transportowa	SWD3075	2	1087
transportowa	RBR9102	2	1218
transportowa	ZSW7961	3	1109
transportowa	TKI0259	3	1329
transportowa	BAU6147	4	1678
transportowa	WGR6362	2	1129
neonatologiczne	TBU6362	3	1349
neonatologiczne	TBU1926	4	1249
podstawowa	TKI1376	2	1248
specjalistyczna	TKI8326	2	1298

Oraz po wykonaniu operacji INSERT:

1	INSERT INTO karetki VALUES (16, 'specjalistyczna', 'RKD7352', 4, '1634');
---	---

specjalistyczna	RKI8326	2	1298
specjalistyczna	RKD7352	4	1634

Testy przebiegły poprawnie.

Dalsze prace nad projektem

Dalsze prace nad projektem mogłyby skupić się na:

- Poprawie interfejsu graficznego aplikacji
- Dodaniu więcej możliwych operacji które będzie można wykonywać na stronie internetowej
- Zmianie wyglądu przycisków i tabel

Wnioski

- Wszystkie z wymagań projektowych zostały spełnione
- Bazy danych są nieodłącznym elementem większości stron internetowych oraz aplikacji
- Interfejs graficzny w znacznym stopniu ułatwia pracę z bazą danych.
- Zastosowanie języka PHP do połączenia z bazą i operacji na niej usprawniło nasze prace nad klientem bazy danych.