

PROJEKT SIECI APTEK

Alicja Wróbel

21.08.2019r.

1. Cel projektu

Celem projektu jest wspólna baza danych dla punktów aptecznych. Ma ona usprawnić działanie aptek oraz optymalizację pracy z dużą ilością danych, co pozwala na zwiększenie efektywności pracowników placówki oraz odciążenie ich od zarządzania dokumentacją.

2. Analiza wymagań

W systemie zapisane są informacje oraz dane kontaktowe o pracownikach, klientach, firmach, produktach, aptekach. Istnieje możliwość zarządzania zamówieniami i dostawami produktów, monitorowania sprzedaży leków względem refundowanych i nier refundowanych, pracownika, klienta, apteki, planowanie i rejestrowanie grafików pracowników, godzin otwarcia aptek.

3. Założenia

- Apteki pracują 8 -22, natomiast w trybie rotacyjnym jedna apteka czynna jest całą dobę,
- Apteki mogą składać zamówienia u wielu dostawców,
- Zamówienia można składać dzięki informacji o dostępnych produktach,
- Pojedyncze zamówienie może dotyczyć wielu produktów,
- Produkty przechowuje się w magazynach,
- Dostawa nie zawsze musi być równa zamówieniu,
- Apteka ma wielu klientów stałych jak i okazjonalnych,
- Klienci mają unikatowe dane personalne, jednak adres zamieszkania może się powielać,
- Klient może zakupić wiele produktów anonimowo, na receptę lub jako stały klient,
- Recepta lub karta stałego klienta zwalnia z kwoty wyznaczonej procentowo,
- NFZ zwraca koszty kupionych leków na receptę,
- W aptekach pracuje 20 farmaceutów oraz 10 kierowników,
- W trakcie zmiany jest od 1-3 farmaceutów oraz 1 kierownik,
- Pracownicy mają ustalony grafik i nie są przydzieleni na stałe do danego punktu aptecznego,
- Można rozliczać pracowników ze względu na sprzedaż,
- Można rozliczać się z dostawcami oraz z NFZ.

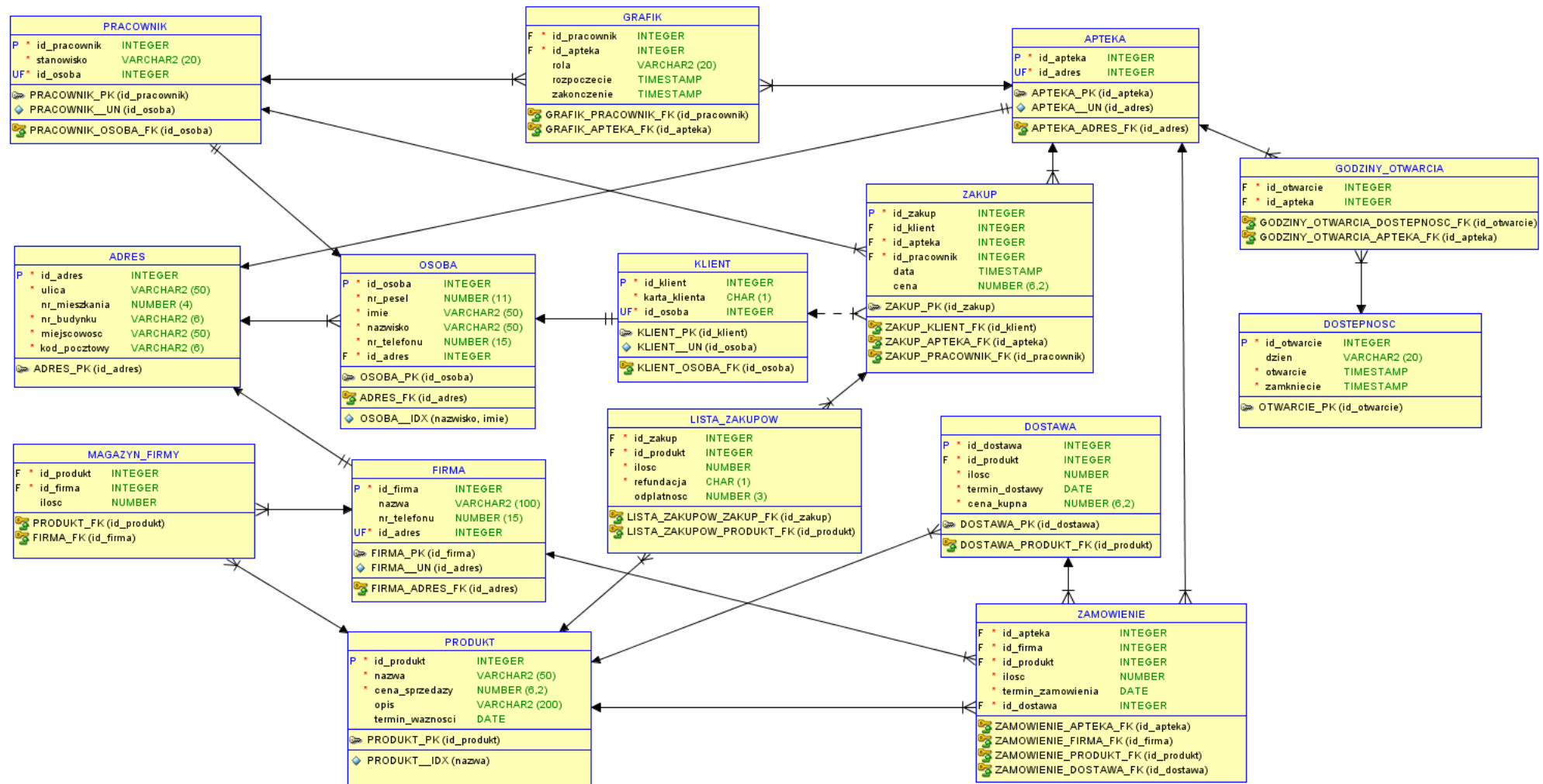
4. Opis logiczny systemu

Baza zawiera listę leków z opisem i ceną sprzedaży. Dany produkt w konkretnych ilościach jest przechowywany w magazynach firm. Firma posiada lokalizacje oraz dane kontaktowe, które pozwalają na złożenie zamówienia. Określić można ilość potrzebnych leków do danego punktu aptecznego o podanej lokalizacji i godzinach otwarcia. W momencie dostarczenia można porównać ilość i rodzaj produktów przywiezionych z zamówionymi. Założyłam, że dostawy odbywają się w wyznaczonym terminie. Apteka zawiera listę wszystkich dostępnych pracowników, która umożliwia planowanie im grafików. Nie są oni przypisani na sztywno do danej placówki, dzięki czemu mamy szerszy zakres wyboru. W celu identyfikacji zatrudnionych, baza zawiera informacje o ich stanowisku, danych personalnych oraz kontaktowych. Farmaceuci oraz kierownicy mogą sprzedawać leki klientom. Klienci pozostają anonimowi jednak, gdy chcą założyć kartę stałego klienta, podają swoje dane personalne, co również umożliwia wystawienie faktury. Ze względu na konieczność podania nr PESEL klienta, zakładam, że każdy klient go posiada.

5. Wykorzystane technologie i narzędzia

Jako system zarządzania bazą danych wykorzystany został Oracle SQL. W celu zaprojektowania modelu użyto Data Modeler oraz do stworzenia bazy danych Oracle SQL Developer i Oracle Database. Część danych została wygenerowana na stronie <https://www.generatedata.com/>.

6. Model fizyczny



7. Implementacja systemu baz danych oraz przykładowych danych

Tabele, relacje oraz ograniczenia zostały wygenerowane automatycznie na podstawie stworzonego modelu fizycznego. Przykład dla tabeli KLIENT.

```
CREATE TABLE klient (
    id_klient      INTEGER NOT NULL,
    karta_klienta  CHAR(1) NOT NULL,
    id_osoba       INTEGER NOT NULL
);

ALTER TABLE klient ADD CONSTRAINT klient_pk PRIMARY KEY ( id_klient );

ALTER TABLE klient ADD CONSTRAINT klient_un UNIQUE ( id_osoba );

ALTER TABLE klient
    ADD CONSTRAINT klient_osoba_fk FOREIGN KEY ( id_osoba )
        REFERENCES osoba ( id_osoba );
```

Dodanie danych do tabeli ZAMÓWIENIE, czyli złożenie zamówień

The screenshot shows a database query tool interface. At the top, there's a 'Worksheet' tab and a 'Query Builder' tab. The main area displays three SQL INSERT statements for the 'ZAMOWIENIE' table. Below the statements, a 'Script Output' window shows the results of the execution, indicating that each statement successfully inserted one row.

```
INSERT INTO ZAMOWIENIE (id_apteka,id_firma,id_produkt,ilosc,termin_zamowienia,id_dostawa) VALUES ('1','2',28,42,'20-08-06',28);
INSERT INTO ZAMOWIENIE (id_apteka,id_firma,id_produkt,ilosc,termin_zamowienia,id_dostawa) VALUES ('1','3',29,52,'20-02-05',29);
INSERT INTO ZAMOWIENIE (id_apteka,id_firma,id_produkt,ilosc,termin_zamowienia,id_dostawa) VALUES ('1','1',30,42,'19-06-07',30);
```

Script Output x

Task completed in 0,067 seconds

1 row inserted.

1 row inserted.

1 row inserted.

8. Mechanizmy przetwarzania danych

Widoki

REJESTR_DOSTAWY_A1 – Przedstawia dostarczone leki do wybranej apteki wraz z ilością i kosztem całkowitym dla danego produktu

```
CREATE VIEW REJESTR_DOSTAWY_A1 AS
SELECT p.id_produkt, SUM(d.ilosc) AS ILOSC, SUM(d.ilosc*d.cena_kupna) AS Koszt
FROM produkt p
JOIN dostawa d ON p.id_produkt = d.id_produkt
JOIN zamowienie z ON z.id_produkt = p.id_produkt
WHERE z.id_apteka = 1
GROUP BY p.id_produkt
ORDER BY p.id_produkt;
```

Script Output x Query Result x
Task completed in 0,059 seconds

ID_PRODUKT	ILOSC	KOSZT
13	63	1134
14	3	105
15	41	615
16	97	1940
17	97	1746
18	58	2030
19	7	105
20	69	1380
21	63	1134
22	53	1855
23	92	1380

ID_PRODUKT	ILOSC	KOSZT
24	64	1280
25	90	1620
26	100	3500
27	20	300
28	42	840
29	52	936
30	42	1470

REJESTR_SPRZEDAZY_A1 – Przedstawia sprzedane leki wraz z ilością, zyskiem przed obliczaniem kosztów i refundacja NFZ

Worksheet		Query Builder	
		<pre>CREATE VIEW REJESTR_SPRZEDAZY_A1 AS SELECT p.id_produkt, SUM(l.ilosc) AS ILOSC, SUM(l.ilosc*p.cena_sprzedazy) AS ZYSK_SPRZEDAZY, SUM(l.ilosc*p.cena_sprzedazy*1.odplatnosc/100) AS ZWROT_NFZ FROM produkt p JOIN lista_zakupow l ON l.id_produkt = p.id_produkt JOIN zakup z ON l.id_zakup = z.id_zakup WHERE z.id_apteka = 1 GROUP BY p.id_produkt ORDER BY p.id_produkt ASC;</pre>	
		<div>Script Output x QueryResult x</div> <div>Task completed in 0,076 seconds</div>	
ID_PRODUKT	ILOSC	ZYSK_SPRZEDAZY	ZWROT_NFZ

1	4	80	18,6
2	4	148	43,29
3	8	128	
4	7	147	
5	7	140	
6	1	37	
7	1	16	
8	1	21	
9	1	20	
10	1	37	
11	3	48	

ID_PRODUKT	ILOSC	ZYSK_SPRZEDAZY	ZWROT_NFZ

12	1	21	
13	1	20	
14	4	148	
15	6	96	
16	5	105	
17	3	60	
18	6	222	
19	2	32	
20	11	231	
21	2	40	
22	1	37	

AKTUALNY_STAN_A1 – Przedstawia aktualny stan dostępnych leków w danej aptece

Worksheet Query Builder

```
CREATE VIEW AKTUALNY_STAN_A1 AS
SELECT vt.id_produkt, vt.ilosc - vr.ilosc AS AKTUALNA_ILOSC FROM REJESTR_DOSTAWY_A1 vt
LEFT JOIN REJESTR_SPRZEDAZY_A1 vr ON vr.id_produkt = vt.id_produkt;
```

Script Output x Query Result x

Task completed in 0,054 seconds

ID_PRODUKT	AKTUALNA_ILOSC
13	62
14	96
15	35
16	92
17	94
18	52
19	5
20	58
21	61
22	52
23	89

ID_PRODUKT	AKTUALNA_ILOSC
24	58
25	89
26	92
27	17
28	36
29	51
30	38

APTEKI – daty i godziny otwarcia aptek z uwzględnieniem trybu rotacyjnego dla dyżurów całodobowych

Worksheet		Query Builder	
		<pre>CREATE VIEW APTEKI AS SELECT a.id_apteka, TO_CHAR(d.otwarcie, 'YYYY-MON-DD HH24:MI') AS OTWARCIE, TO_CHAR(d.zamkniecie, 'YYYY-MON-DD HH24:MI') AS ZAMKNIECIE FROM apteka a JOIN godziny_otwarcia g ON g.id_apteka = a.id_apteka JOIN dostepnosc d ON d.id_otwarcie = g.id_otwarcie ORDER BY d.otwarcie, a.id_apteka; SELECT * FROM APTEKI;</pre>	
		Script Output x	
		Task completed in 0,025 seconds	
ID_APTEKA	OTWARCIE	ZAMKNIECIE	
1	2019-STY-01 08:00	2019-STY-02 22:00	
2	2019-STY-01 08:00	2019-STY-01 22:00	
3	2019-STY-01 08:00	2019-STY-01 22:00	
4	2019-STY-01 08:00	2019-STY-01 22:00	
2	2019-STY-02 08:00	2019-STY-03 22:00	
3	2019-STY-02 08:00	2019-STY-02 22:00	
4	2019-STY-02 08:00	2019-STY-02 22:00	
1	2019-STY-03 08:00	2019-STY-04 22:00	
2	2019-STY-03 08:00	2019-STY-03 22:00	
4	2019-STY-03 08:00	2019-STY-03 22:00	
1	2019-STY-04 08:00	2019-STY-05 22:00	
ID_APTEKA	OTWARCIE	ZAMKNIECIE	
2	2019-STY-04 08:00	2019-STY-04 22:00	
3	2019-STY-04 08:00	2019-STY-04 22:00	
2	2019-STY-05 08:00	2019-STY-06 22:00	
3	2019-STY-05 08:00	2019-STY-05 22:00	
4	2019-STY-05 08:00	2019-STY-05 22:00	
1	2019-STY-06 08:00	2019-STY-07 22:00	
3	2019-STY-06 08:00	2019-STY-06 22:00	
4	2019-STY-06 08:00	2019-STY-06 22:00	
1	2019-STY-07 08:00	2019-STY-08 22:00	
2	2019-STY-07 08:00	2019-STY-07 22:00	

PRACOWNICY – grafik pracowników dla danego punktu aptecznego z uwzględnieniem założeń dotyczących ilości pracowników na zmianie

Worksheet

Query Builder

CREATE VIEW PRACOWNICY AS

SELECT g.id_pracownik, p.stanowisko, TO_CHAR(g.rozpozecie, 'YYYY-MON-DD HH24:MI') AS ROZPOCZECIE,

TO_CHAR(g.zakonczenie, 'YYYY-MON-DD HH24:MI') AS ZAKONCZENIE

FROM grafik g

JOIN pracownik p ON g.id_pracownik = p.id_pracownik

WHERE g.id_apteka = 1

ORDER BY g.rozpozecie;

SELECT * FROM PRACOWNICY;

Script Output x

Task completed in 0,041 seconds

ID_PRACOWNIK	STANOWISKO	ROZPOCZECIE	ZAKONCZENIE
21	Kierownik	2019-STY-01 08:00	2019-STY-01 16:00
1	Farmaceuta	2019-STY-01 08:00	2019-STY-01 16:00
3	Farmaceuta	2019-STY-01 14:00	2019-STY-01 22:00
2	Farmaceuta	2019-STY-01 14:00	2019-STY-01 22:00
22	Kierownik	2019-STY-01 16:00	2019-STY-02 00:00
4	Farmaceuta	2019-STY-01 22:00	2019-STY-02 06:00
23	Kierownik	2019-STY-02 00:00	2019-STY-02 08:00
1	Farmaceuta	2019-STY-02 06:00	2019-STY-02 14:00
21	Kierownik	2019-STY-02 08:00	2019-STY-02 16:00
2	Farmaceuta	2019-STY-02 14:00	2019-STY-02 22:00
3	Farmaceuta	2019-STY-02 14:00	2019-STY-02 22:00
ID_PRACOWNIK	STANOWISKO	ROZPOCZECIE	ZAKONCZENIE
22	Kierownik	2019-STY-02 16:00	2019-STY-02 22:00
21	Kierownik	2019-STY-03 08:00	2019-STY-03 16:00
1	Farmaceuta	2019-STY-03 08:00	2019-STY-03 16:00
2	Farmaceuta	2019-STY-03 14:00	2019-STY-03 22:00
3	Farmaceuta	2019-STY-03 14:00	2019-STY-03 22:00
22	Kierownik	2019-STY-03 16:00	2019-STY-03 22:00
21	Kierownik	2019-STY-04 08:00	2019-STY-04 16:00
1	Farmaceuta	2019-STY-04 08:00	2019-STY-04 16:00
2	Farmaceuta	2019-STY-04 14:00	2019-STY-04 22:00
3	Farmaceuta	2019-STY-04 14:00	2019-STY-04 22:00
22	Kierownik	2019-STY-04 16:00	2019-STY-04 22:00

RAPORT_PRACOWNIK – wykaz sprzedaży danego pracownika

The screenshot shows a database Query Builder window. The top pane contains SQL code for creating a view and selecting data. The bottom pane shows the results of the query in a table format.

```
CREATE VIEW RAPORT_PRACOWNIK AS
SELECT l.id_produkt, l.ilosc AS ILOSC, z.cena/l.ilosc AS CENA, z.cena AS SUMA FROM zakup z
JOIN pracownik p ON p.id_pracownik = z.id_pracownik
JOIN lista_zakupow l ON l.id_zakup = z.id_zakup
WHERE p.id_pracownik = 1 group by l.id_produkt, z.cena, l.ilosc
ORDER BY l.id_produkt ASC;

SELECT * FROM RAPORT_PRACOWNIK;
```

ID_PRODUKT	ILOSC	CENA	SUMA
1	1	21,47	21,47
2	1	21,47	21,47
3	2	45,21	90,42
4	1	40,84	40,84
5	1	40,84	40,84

Indeksy

Zostały utworzone indeksy do tabeli PRODUKT na kolumnie nazwa oraz do tabeli OSOBA na kolumnach nazwisko oraz imię w celu szybkiego wyszukiwania.

The screenshot shows a database Query Builder window with SQL code for creating two indexes.

```
CREATE INDEX produkt_idx ON
produkt ( nazwa ASC );

CREATE INDEX osoba_idx ON
osoba ( nazwisko ASC, imie ASC );
```

9. Przykładowe testy

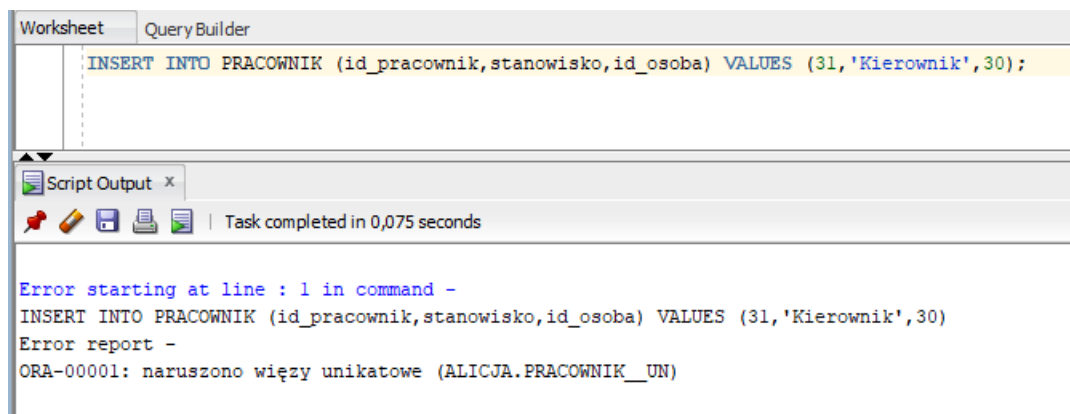
Not null

The screenshot shows a database Query Builder window with an SQL insert statement. Below the statement, the script output shows an error message.

```
INSERT INTO KLIENT (id_klient,id_osoba) VALUES (30,60);
```

Error starting at line : 1 in command -
INSERT INTO KLIENT (id_klient,id_osoba) VALUES (30,60)
Error report -
ORA-01400: nie można wstawić wartości NULL do ("ALICJA"."KLIENT"."KARTA_KLIENTA")

Unique



The screenshot shows a database query tool interface. At the top, there are two tabs: 'Worksheet' and 'Query Builder'. The 'Query Builder' tab is active, displaying an SQL INSERT statement: `INSERT INTO PRACOWNIK (id_pracownik, stanowisko, id_osoba) VALUES (31, 'Kierownik', 30);`. Below the query editor, there is a 'Script Output' window. It shows a status bar indicating 'Task completed in 0,075 seconds'. Below that, an error message is displayed: 'Error starting at line : 1 in command -', followed by the SQL statement, and then 'Error report -' and the specific error: 'ORA-00001: naruszono więzy unikatowe (ALICJA.PRACOWNIK__UN)'.

```
Worksheet Query Builder
INSERT INTO PRACOWNIK (id_pracownik, stanowisko, id_osoba) VALUES (31, 'Kierownik', 30);

Script Output x
Task completed in 0,075 seconds

Error starting at line : 1 in command -
INSERT INTO PRACOWNIK (id_pracownik, stanowisko, id_osoba) VALUES (31, 'Kierownik', 30)
Error report -
ORA-00001: naruszono więzy unikatowe (ALICJA.PRACOWNIK__UN)
```

10. Przyszły rozwój bazy danych

W trakcie projektu zauważyłam dalszą możliwość jego rozbudowy. W punktach aptecznych są dwa stanowiska, można by było wydzielić je ze względu na odpowiedzialność pełnionej funkcji. Należałoby stworzyć użytkowników Farmaceuta oraz Kierownik, a następnie nadać odpowiednia uprawnienia. Dla przykładu, dostęp do ustalania grafików ma tylko kierownik, natomiast do zakładania kart stałego klienta każdy pracownik. Oprócz tego dodatkową funkcją mogłaby być tabela logów, do której informacje zapisywane byłby dzięki wyzwalaczowi na tabeli zamówień. Posiadałaby ona informacje o osobie składającej zamówienie oraz przeprowadzonej zmianie (usunięcie, dodanie, modyfikacja).