

# Sprawozdanie z projektu podsumującego MySQL z Baz Danych

## Klaudia Balicka, Anna Czesak, Wioletta Drąg,

### gr. projektowa 1 IT, WIMiP

## Projekt MySQL& PHP

### Sklep z ubraniami

#### Idea projektu

Głównym celem projektu było stworzenie bazy danych, która mogłaby być używana w sklepie z ubraniami.

Projekt ten ma spełniać podstawowe funkcje, które są potrzebne w zarządzaniu danymi w sklepie, między innymi: przechowywanie danych o produktach znajdujących się na stanie, o klientach oraz ich adresach potrzebnych do wysyłki produktów oraz dane o złożonych zamówieniach.

Prosty interfejs użytkownika utworzony w języku php umożliwia łatwe poruszanie się między funkcjonalnościami programu.

Funkcjonalności te spełniają założenia CRUD i pozwalają na proste i przejrzyste poruszanie się między danymi oraz na ich edycję.

Tworząc elementy projektu korzystaliśmy z DQL - Data Query Language, DML – Data Manipulation Language, DDL – Data Definition Language oraz DCL – Data Control Language

W bazie danych o nazwie sklep.sql przechowywane są tabele:

- Klienci (id, Imię, nazwisko, telefon, mail, id adresu )

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
1	id	int(11)			Nie	Brak			Zmień  Usuń  Więcej
2	imie	varchar(30)	utf8_polish_ci		Nie	Brak			Zmień  Usuń  Więcej
3	nazwisko	varchar(30)	utf8_polish_ci		Nie	Brak			Zmień  Usuń  Więcej
4	nr_telefonu	int(11)			Nie	Brak			Zmień  Usuń  Więcej
5	mail	varchar(30)	utf8_polish_ci		Nie	Brak			Zmień  Usuń  Więcej
6	id_adresu	int(11)			Nie	Brak			Zmień  Usuń  Więcej

- Adresy (id, Miasto, ulica, dom/lokal)

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
1	id	int(11)			Nie	Brak			Zmień  Usuń  Więcej
2	ulica	varchar(30)	utf8_polish_ci		Nie	Brak			Zmień  Usuń  Więcej
3	dom_lokal	varchar(20)	utf8_polish_ci		Nie	Brak			Zmień  Usuń  Więcej
4	miasto	varchar(20)	utf8_polish_ci		Nie	Brak			Zmień  Usuń  Więcej

- Zamowienia (id, numer id klienta, id produktu, ilość)

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
1	id	int(11)			Nie	Brak			Zmień  Usuń  Więcej
2	id_klienta	int(11)			Nie	Brak			Zmień  Usuń  Więcej
3	id_produktu	int(11)			Nie	Brak			Zmień  Usuń  Więcej
4	ilosc	int(11)			Nie	Brak			Zmień  Usuń  Więcej

- Produkty (id, Nazwa, cena, kolor , rozmiar, id kategorii, dostępność(ilość produktu))

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
<input type="checkbox"/>	1 id	int(11)			Nie	Brak			Zmień  Usuń  Więcej
<input type="checkbox"/>	2 nazwa	varchar(30) utf8_polish_ci			Tak	NULL			Zmień  Usuń  Więcej
<input type="checkbox"/>	3 cena	float			Nie	Brak			Zmień  Usuń  Więcej
<input type="checkbox"/>	4 kolor	varchar(20) utf8_polish_ci			Nie	Brak			Zmień  Usuń  Więcej
<input type="checkbox"/>	5 rozmiar	varchar(5) utf8_polish_ci			Nie	Brak			Zmień  Usuń  Więcej
<input type="checkbox"/>	6 id_kategorii	int(11)			Nie	Brak			Zmień  Usuń  Więcej
<input type="checkbox"/>	7 dostepnosc	int(11)			Nie	Brak			Zmień  Usuń  Więcej

- Nowa\_kolekcja\_produktyw(id, nazwa, cena, kolor, rozmiar, id\_kategorii, dostepnosc)

				id	nazwa	cena	kolor	rozmiar	id_kategorii	dostepnosc
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	1	spodnie	50	zielony	M	0	2
<input type="checkbox"/>	Edytuj	Kopiuj	Usuń	2	skarpetki	12	niebieski	XS	2	10

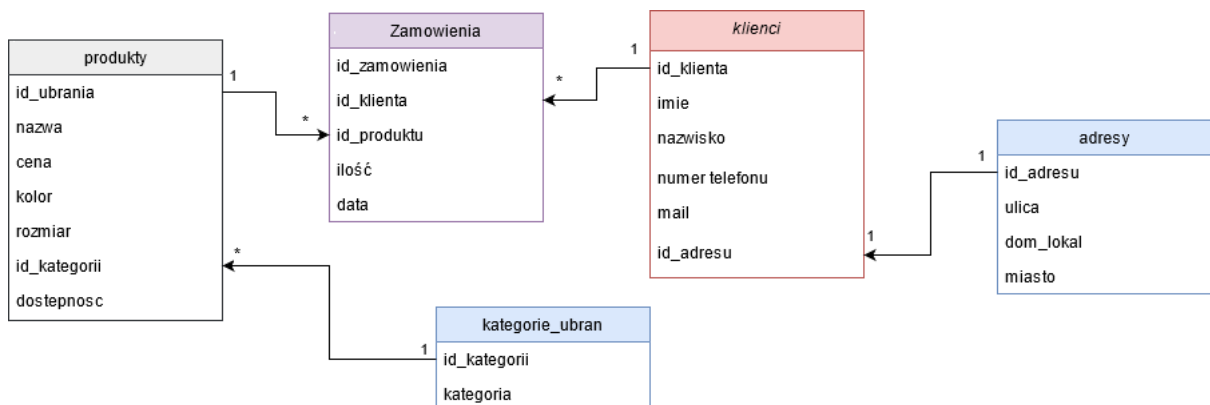
- Kategorie\_produktyw(id, kategoria)

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
<input type="checkbox"/>	1 id	int(11)			Nie	Brak			Zmień  Usuń  Więcej
<input type="checkbox"/>	2 kategoria	varchar(20) utf8_polish_ci			Nie	Brak			Zmień  Usuń  Więcej

- Produkty\_zmiany (id, data zmiany, polecenie, poprzednia ilość, nowa ilość (produktów))

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
1	ID	int(11)			Nie	Brak		AUTO_INCREMENT	Zmień  Usuń  Więcej
2	data_zmiany	date			Nie	Brak			Zmień  Usuń  Więcej
3	polecenie	varchar(20) utf8mb4_general_ci			Nie	Brak			Zmień  Usuń  Więcej
4	poprzednia_ilosc	int(11)			Nie	Brak			Zmień  Usuń  Więcej
5	nowa_ilosc	int(11)			Nie	Brak			Zmień  Usuń  Więcej

## DIAGRAM ERD z uzasadnieniem relacji



Tabele adresy i klienci połączone są relacją jeden do jeden, w tabeli klienci znajduje się klucz obcy id\_adresu który jest kluczem podstawowym w tabeli adresy.

Tabela z zamówieniami zawiera w sobie id\_produkту, który jest zamawiany oraz id\_klienta, który składa zamówienie. Są to relacje jeden do wielu (nieograniczona ilość klientów może złożyć zamówienie oraz nieograniczona liczba produktów może być zamówiona).

Tabela produkty zawiera kolumnę id\_kategorii, jest to klucz obcy do tabeli kategorie\_ubrań.

## ZŁĄCZENIA POZIOME

- Przykładowym złączeniem poziomym jest złączenie tabel klienci oraz adresy za pomocą INNER JOIN:

Akurat w tym przypadku wykorzystaliśmy to złączenie do utworzenia widoku:

```
MySQL localhost:3306 sklep SQL > CREATE OR REPLACE VIEW widok_klient_adres AS SELECT id_adresu, imie, nazwisko, nr_telefonu, mail, ulica, dom_lokal, miasto FROM adresy INNER JOIN klienci ON adresy.id=klienci.id_adresu;
Query OK, 0 rows affected (0.0253 sec)
MySQL localhost:3306 sklep SQL > select * from widok_klient_adres;
```

id_adresu	imie	nazwisko	nr_telefonu	mail	ulica	dom_lokal	miasto
0	Alicja	Nowak	123456789	nowak@wp.pl	Zielona	13a/6	Kraków
1	Jan	Kowalski	321432543	kowal@wp.pl	Komorowskiego	13	Kraków
2	Anna	Kusnierz	987435678	ank@onet.pl	Powstańców	44/9	Kraków
3	Mateusz	Fak	124907544	fak@gmail.com	Zwyciestwa	15	Kraków

4 rows in set (0.0009 sec)

Wykorzystanie:

## BAZA KLIENTÓW

ID KLIENTA	IMIĘ	NAZWISKO	TELEFON	EMAIL	ULICA	DOM/LOKAL	MIASTO
0	Alicja	Nowak	123456789	nowak@wp.pl	Zielona	13a/6	Kraków
1	Jan	Kowalski	321432543	kowal@wp.pl	Komorowskiego	13	Kraków
2	Anna	Kusnierz	987435678	ank@onet.pl	Powstańców	44/9	Kraków

- Kolejne złączenie użyte zostało przy wyświetlaniu produktów. Tym razem łączymy dane z tabeli produkty oraz kategorie:

```
$wynik="SELECT * FROM kategorie_ubran INNER JOIN produkty ON kategorie_ubran.id=produkty.id_kategorii";
```

Wykorzystanie:

## BAZA PRODUKTÓW

ID	UBRANIA	NAZWA	CENA	KOLOR	ROZMIAR	KATEGORIA	DOSTĘPNOŚĆ
0		mom jeans	105	niebieski	M	spodnie	5
1		sukienka	60	niebieski	S	sukienka	3
2		skarpetki	19	różowy	S	bielizna	8

### ZŁĄCZENIA PIONOWE

- Przykładowym złączeniem pionowym jest złączenie tabel produkty oraz produkty nowa kolekcja za pomocą UNION:

```
MySQL localhost:3306 sklep SQL > SELECT id, nazwa, cena, kolor, rozmiar, id_kategorii, dostepnosc FROM produkty
UNION ALL SELECT id, nazwa, cena, kolor, rozmiar, id_kategorii, dostepnosc FROM nowa_kolekcja_produkty;
+-----+-----+-----+-----+-----+-----+-----+
| id | nazwa | cena | kolor | rozmiar | id_kategorii | dostepnosc |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | koszula | 19 | biały | S | 1 | 6 |
| 2 | Sukienka | 64 | różowy | XS | 3 | 9 |
| 1 | spodnie | 50 | zielony | M | 0 | 2 |
| 2 | skarpetki | 12 | niebieski | XS | 2 | 10 |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.0006 sec)
```

```
SELECT id, nazwa, cena, kolor, rozmiar, id_kategorii, dostepnosc FROM produkty
UNION SELECT id, nazwa, cena, kolor, rozmiar, id_kategorii, dostepnosc FROM
nowa_kolekcja_produkty;
```

Wykorzystanie:

## BAZA PRODUKTÓW

NAZWA	CENA	KOLOR	ROZMIAR	DOSTĘPNOŚĆ
koszula	19	biały	S	6
Sukienka	64	różowy	XS	9
spodnie	50	zielony	M	2
skarpetki	12	niebieski	XS	10

[Powrót](#)

### PROCEDURY

- dostępność produktów – procedura wyświetla produkty, których dostępność jest mniejsza niż podana przez użytkownika w formularzu liczba.

```
MySQL localhost:3306 sklep SQL > DELIMITER //
MySQL localhost:3306 sklep SQL > CREATE PROCEDURE dostepnosc_produkty (IN min_dostepnosc INT(11))
-> BEGIN SELECT * FROM produkty WHERE dostepnosc < min_dostepnosc;
-> END //
Query OK, 0 rows affected (0.0127 sec)
```

Wykorzystanie:

## DOSTEPNOSC PRODUKTÓW

Wyswietl produkty, których jest mniej niz:

ID	UBRANIA	NAZWA	CENA	KOLOR	ROZMIAR	DOSTĘPNOŚĆ
----	---------	-------	------	-------	---------	------------

1		koszula	45	biały	S	4
---	--	---------	----	-------	---	---

[Powrót](#)

- Zmiana ceny produktów - aktualizacja

ceny produktu na cenę wpisaną przez użytkownika do formularza

```
MySQL localhost:3306 sklep SQL > DELIMITER //
MySQL localhost:3306 sklep SQL > CREATE PROCEDURE ustaw_cena (IN nowa_cena FLOAT, id_p INT(11))
    -> BEGIN
    -> UPDATE produkty SET cena=nowa_cena WHERE id=id_p; END //
Query OK, 0 rows affected (0.0059 sec)
```

Wykorzystanie:

## EDYCJA CENY PRODUKTÓW

id produktu, którego cenę chcesz zmienić:

nowa cena:

Zaktualizowano cene produktu: 2

[Powrót](#)

## WYZWALACZE

- Aktualizacja produktu

Utworzyliśmy wyzwalacz,

```
MySQL localhost:3306 sklep SQL > DELIMITER //
MySQL localhost:3306 sklep SQL > CREATE OR REPLACE TRIGGER aktualizacja_produktu AFTER UPDATE ON produkty FOR EACH ROW BEGIN INSERT INTO produkty_zmiany (ID, data_zmiany, polecenie, poprzednia_ilosc, nowa_ilosc) VALUES (NEW.ID, CURDATE(), 'UPDATE', OLD.dostepnosc, NEW.dostepnosc);
--> END //
Query OK, 0 rows affected (0.0627 sec)
```

który zapisuje zmiany dokonane w nowoutworzonej tabeli produkty\_zmiany

```
MySQL localhost:3306 sklep SQL > CREATE TABLE produkty_zmiany (ID INT(11) NOT NULL AUTO_INCREMENT, data_zmiany DATE NOT NULL, polecenie VARCHAR(20) NOT NULL, poprzednia_ilosc INT(11) NOT NULL, nowa_ilosc INT(11) NOT NULL, PRIMARY KEY (ID));
Query OK, 0 rows affected (0.0178 sec)
MySQL localhost:3306 sklep SQL > describe produkty_zmiany;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID | int(11) | NO | PRI | NULL | auto_increment |
| data_zmiany | date | NO | | NULL | |
| polecenie | varchar(20) | NO | | NULL | |
| poprzednia_ilosc | int(11) | NO | | NULL | |
| nowa_ilosc | int(11) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.0081 sec)
```

Wywołuje się on po instrukcji UPDATE

```
MySQL localhost:3306 sklep SQL > UPDATE produkty SET dostepnosc=6 WHERE ID=1;
Query OK, 1 row affected (0.0101 sec)

Rows matched: 1 Changed: 1 Warnings: 0
MySQL localhost:3306 sklep SQL > SELECT * FROM produkty_zmiany;
+-----+-----+-----+-----+-----+
| ID | data_zmiany | polecenie | poprzednia_ilosc | nowa_ilosc |
+-----+-----+-----+-----+-----+
| 1 | 2021-05-17 | UPDATE | 4 | 6 |
+-----+-----+-----+-----+-----+
1 row in set (0.0006 sec)
```

## WIDOKI

- Widok łączy w sobie pola tabeli adresy oraz klienci (złączenie tabel). Użycie w pliku wyswietl\_klientow.php

```
MySQL localhost:3306 sklep SQL > CREATE OR REPLACE VIEW widok_klient_adres AS SELECT id_adresu, imie, nazwisko, nr_telefonu, mail, ulica, dom_lokal, miasto FROM adresy INNER JOIN klienci ON adresy.id=klienci.id_adresu;
Query OK, 0 rows affected (0.0253 sec)
MySQL localhost:3306 sklep SQL > select * from widok_klient_adres;
+-----+-----+-----+-----+-----+-----+-----+
| id_adresu | imie | nazwisko | nr_telefonu | mail | ulica | dom_lokal | miasto |
+-----+-----+-----+-----+-----+-----+-----+
| 0 | Alicja | Nowak | 123456789 | nowak@wp.pl | Zielona | 13a/6 | Kraków |
| 1 | Jan | Kowalski | 321432543 | kowal@wp.pl | Komorowskiego | 13 | Kraków |
| 2 | Anna | Kusnierz | 987435678 | ank@onet.pl | Powstańców | 44/9 | Kraków |
| 3 | Mateusz | Fąk | 124907544 | fak@gmail.com | Zwyciestwa | 15 | Kraków |
+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.0009 sec)
```

```
$wynik="SELECT * FROM kategorie_ubran INNER JOIN produkty ON
kategorie_ubran.id=produkty.id_kategorii";
```

Wykorzystanie:

## BAZA KLIENTÓW

ID KLIENTA	IMIĘ NAZWISKO	TELEFON	EMAIL	ULICA	DOM/LOKAL	MIASTO
0	Alicja Nowak	123456789	nowak@wp.pl	Zielona	13a/6	Kraków
1	Jan Kowalski	321432543	kowal@wp.pl	Komorowskiego 13		Kraków
2	Anna Kusnierz	987435678	ank@onet.pl	Powstańców	44/9	Kraków

- Widok wyświetlający nazwę produktu i jego cenę

```
MySQL localhost:3306 sklep SQL > CREATE VIEW widok_produkt_cena AS SELECT nazwa, cena FROM produkty;
Query OK, 0 rows affected (0.0043 sec)
MySQL localhost:3306 sklep SQL > select * from widok_produkt_cena;
+-----+-----+
| nazwa | cena |
+-----+-----+
| mom jeans | 80 |
| koszula | 50 |
| Sukienka | 40 |
+-----+-----+
3 rows in set (0.0010 sec)
```

## Aktualne ceny produktów

nazwa	cena
mom jeans	80
koszula	45
Sukienka	40

[Powrót](#)

### CRUD:

**CREATE** - utworzenie oraz dodawanie nowych informacji

**READ** - wyświetlanie oraz odczytywanie istniejących informacji

**UPDATE** - modyfikowanie oraz edytowanie istniejących informacji

**DELETE** - usuwanie istniejących informacji

CREATE

Dodano klienta!

## Dodaj do bazy klientow

### DANE OSOBISTE

id klienta:

imię:

nazwisko:

e-mail:

numer telefonu:

### ADRES

ulica:

dom/lokal:

miasto:

## Dodaj do bazy produktow

### DANE OSOBISTE

id produktu:

nazwa:

cena:

kolor:

rozmiar:

kategoria:

dostepnosc:

[Powrót](#)



# Dodaj do zamówień

id zamówienia:

5

id klienta:

3

id produktu:

2

ilosc:

1

dodaj

[Powrót](#)

READ

# BAZA KLIENTÓW

ID KLIENTA	IMIE	NAZWISKO	TELEFON	EMAIL	ULICA	DOM/LOKAL	MIASTO
0	Alicja	Nowak	123456789	nowak@wp.pl	Zielona	13a/6	Kraków
1	Jan	Kowalski	321432543	kowal@wp.pl	Komorowskiego 13		Kraków
2	Anna	Kusnierz	987435678	ank@onet.pl	Powstańców	44/9	Kraków

# BAZA PRODUKTÓW

ID UBRANIA	NAZWA	CENA	KOLOR	ROZMIAR	KATEGORIA	DOSTEPNOŚĆ
0	mom jeans	129	niebieski	M	spodnie	5
1	koszula	19	biały	S	bluzka	6
2	Sukienka	100	różowy	XS	sukienka	9

[Powrót](#)  
[Sprawdz dostepnosc produktow](#)

# BAZA ZAMÓWIEŃ

NR KLIENT PRODUKTU ILOSC

0	2	koszula	1
1	1	mom jeans	1
2	0	koszula	1
3	3	koszula	1

[Powrót](#)

UPDATE

## EDYCJA CENY PRODUKTÓW

id produktu, którego cenę chcesz zmienić:

nowa cena:

aktualizuj

Zaktualizowano cene produktu: 0

[Powrót](#)

Zaktualizowano produkt: 2

## Obniż cene produktu

Ustaw wielkość zniżki [%]:

Id produktu podlegającego zniżce:

aktualizuj

[Zobacz aktualne ceny](#)

[Powrót](#)

## DELETE

Usunięto produkt: 2

## Usuń produkt z bazy danych

id\_klienta:

[Powrót](#)

### PODSUMOWANIE

MySQL jest najbardziej popularnym open sourceowym systemem zarządzania relacyjną bazą danych. Jest on częścią popularnego pakietu oprogramowania służącego do rozwoju aplikacji webowych zwanego LAMP czyli Linux, Apache, MySQL oraz PHP.

MySQL będąc oprogramowaniem typu open source oferuje szeroki zestaw możliwości, które są nieustannie usprawnianie.

Możliwość tworzenia widoków zapewnia, że wrażliwe dane nie dostaną się w niepowołane ręce. Tworzenie oraz używanie wyzwalaczy daje automatycznie wykonanie procedur, które wykonują się w reakcji na pewne zdarzenia w tabeli danych.

CRUD realizuje podstawowe operacje implementowane w aplikacjach bazodanowych. Jest stosowany najczęściej w odniesieniu do interfejsu użytkownika w większości aplikacjach, które pozwalają użytkownikowi na:

- utworzenie oraz dodawanie nowych informacji
- wyświetlanie oraz odczytywanie istniejących informacji
- modyfikowanie oraz edytowanie istniejących informacji
- usuwanie istniejących informacji

Technologie, takie jak PHP, umożliwiają tworzenie dynamicznych serwisów WWW, które coraz częściej są połączone z bazami danych z MySQL. Bezpłatny dostęp do tej technologii, jej potencjał i wydajność przyczyniły się do niesamowitej popularności wśród webmasterów.

### Link do repozytorium GitHub z projektem:

<https://github.com/Anna->

[Czesak/Sklep?fbclid=IwAR02yQGDZ6ZJ8ze1T5vC65eFj\\_\\_qbVz8VpqBND9Ma9zvb0QTEz29qyV4rnE](https://github.com/Anna-Czesak/Sklep?fbclid=IwAR02yQGDZ6ZJ8ze1T5vC65eFj__qbVz8VpqBND9Ma9zvb0QTEz29qyV4rnE)