

PWSZ – Nowy Sącz	
Zaawansowane technologie bazodanowe – 2019/2020	
Temat projektu: Bazodanowy system komputerowy realizowany w oparciu o motor MySQL oraz język PHP.	
Imię i nazwisko: Jakub Węglarz	Grupa: Grupa P2
Data: 16.01.2020	Ocena:

Elementy projektu:

1. Strona tytułowa.
2. Spis treści.
3. Opis rzeczywistego problemu dla którego należy zaprojektować system (minimum pięć encji) zgodnie z wybranym tematem.
4. Normalizacja bazy danych.
5. Schemat bazy danych w formie diagramu klas.
6. Opis wybranego motoru bazodanowego, oraz funkcji bazodanowych wybranego języka programowania.
7. Należy wdrożyć ograniczniki, związki, widoki, procedury składowe, wyzwalacze, transakcje.
8. Interfejs użytkownika umożliwiający podstawowe operacje z zakresu DQL+DML.
9. Opis wdrożenia, oraz przeprowadzone testy działania systemu.
10. Wnioski.
11. Literatura.

Spis treści

Opis problemu	3
User Story	3
Normalizacja bazy danych	4
Schemat bazy danych (diagram klas)	4
Opis motoru bazodanowego i funkcji bazodanowych języka php	5
Opis motoru.....	5
Opis funkcji bazodanowych języka PHP	5
Przykłady wdrożenia wymaganych elementów	7
Ograniczniki	7
Związki	7
Widoki	7
Procedury	7
Wyzwalacze	8
Transakcje.....	8
Interfejs użytkownika	9
Opis wdrożenia i testy	10
Możliwości działania aplikacji.....	11
Wnioski	13
Literatura	14

Opis problemu

Na potrzeby projektu zbudowany został system bazodanowy do obsługi hurtowni sprzętu komputerowego. System składa się z bazy danych MySQL oraz aplikacji napisanej w języku PHP (logika) oraz HTML+CSS (interfejs). Poniżej zostały przedstawione tzw. User Story w których użytkownik określa wymagania w odniesieniu do aplikacji i operacji możliwych do zrealizowania na bazie danych.

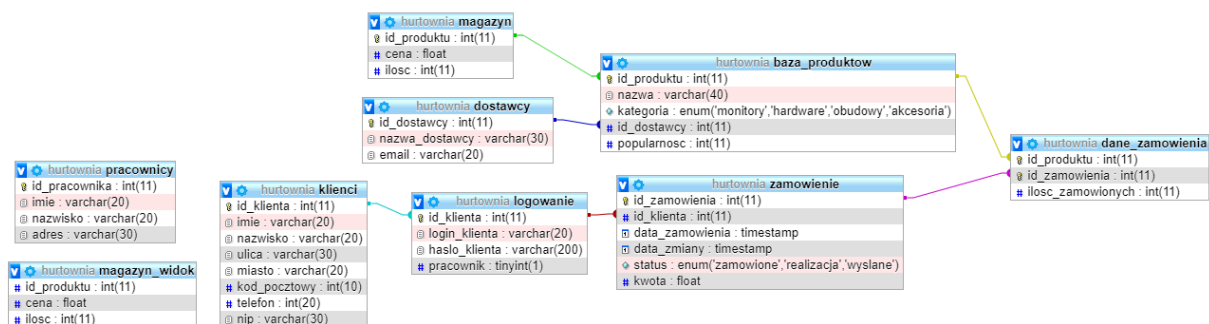
User Story

1. Jako klient chcę mieć możliwość zalogowania na konto klienta wcześniej utworzone przez administratora w celu korzystania z funkcji programu.
2. Jako klient chcę mieć możliwość składania zamówień w celu współpracy z hurtownią.
3. Jako klient chcę mieć możliwość podglądu zamówienia w celu weryfikacji poprawności złożonego zamówienia.
4. Jako klient chcę mieć możliwość pobierania faktur aby wygodnie rozliczać zakupy.
5. Jako klient chcę mieć możliwość podglądu dostępnych produktów aby zapoznać się z nowościami.
6. Jako klient chcę mieć możliwość wyszukiwania produktów aby szybciej znajdować interesujące mnie rzeczy.
7. Jako klient chcę mieć możliwość sprawdzania stanu realizacji zamówień aby kontrolować terminy dostaw.
8. Jako klient chcę mieć możliwość dodawania i edycji danych osobowych w celu płynnego kontaktu z obsługą sklepu.
9. Jako pracownik chcę mieć możliwość logowania się do konta pracowniczego wcześniej utworzonego przez administratora w celu obsługi zamówień.
10. Jako pracownik chcę mieć możliwość realizacji zamówień oraz/lub zmiany ich stanu w celu informowania klientów na bieżąco o postępie.
11. Jako pracownik chcę mieć możliwość automatycznego generowania faktur do zamówień w celu łatwego rozliczania klientów.
12. Jako pracownik chcę mieć możliwość sprawdzania stanu magazynu aby wiedzieć jaki towar ewentualnie domówić.

Normalizacja bazy danych

Jest to proces mający na celu eliminację powtarzających się danych w relacyjnej bazie danych. Główna idea polega na trzymaniu danych w jednym miejscu, a w razie potrzeby linkowania do danych. Taki sposób tworzenia bazy danych zwiększa bezpieczeństwo danych i zmniejsza ryzyko powstania niespójności (w szczególności problemów anomalii). Istnieją sposoby ustalenia czy dany schemat bazy danych jest „znormalizowany”, a jeżeli jest to jak bardzo. Jednym ze sposobów jest przyrównanie danej bazy do schematów zwanych postaciami normalnymi (ang. normal forms lub NF). Normalizacja bazy danych do konkretnej postaci może wymagać rozbicia dużych tabel na mniejsze i przy każdym wykonywaniu zapytania do bazy danych ponownego ich łączenia. Zmniejsza to wydajność, więc w niektórych przypadkach świadoma denormalizacja (stan bez normalizacji) jest lepsza – zwłaszcza w systemach niekorzystających z modelu relacyjnego (np. OLAP). Normalizacja nie usuwa danych, tylko zmienia schemat bazy danych. Normalizacja przeprowadza bazę danych z jednego stanu spójnego (przed normalizacją) w inny stan spójny (po normalizacji). Jedyna różnica polega na innym układzie danych i relacji pomiędzy nimi, ale bez utraty danych (ewentualnie dodawane są nowe klucze główne).

Schemat bazy danych (diagram klas)



Opis motoru bazodanowego i funkcji bazodanowych języka php

Opis motoru

InnoDB – mechanizm (silnik, ang. engine) składowania danych, zaprojektowany przez fińską firmę Innobase Oy, która została przejęta w 2006 przez koncern Oracle Corporation. Dystrybuowany jest głównie na licencji GNU General Public License. Jest obecny standardowo we wszystkich nowych wersjach MySQL, a od wersji 5.5 jest dla MySQL mechanizmem domyślnym. Zastosowanie InnoDB umożliwia korzystanie z takich funkcji bazodanowych jak transakcje i klucze obce. Jest też zgodny ze standardem ACID. W silniku tym dostępne są dwa sposoby magazynowania danych: plik lub grupa plików wspólne dla wszystkich baz i tabel, lub też po jednym pliku z danymi dla każdej tabeli z osobna. Inne ważne cechy InnoDB to: blokady na poziomie wierszy, możliwość kompresji danych, oraz MVCC.

Opis funkcji bazodanowych języka PHP

`mysql_affected_rows` - Zwraca liczbę wierszy przetworzonych w poprzedniej operacji MySQL

`mysql_client_encoding` - Zwraca nazwę zestawu znaków

`mysql_close` - Zamyka połączenie MySQL

`mysql_connect` - Otwiera połączenie do serwera MySQL

`mysql_create_db` - Tworzy bazę MySQL

`mysql_data_seek` - Przesuwa wewnętrzny wskaźnik wyniku

`mysql_db_name` - Pobiera dane wynikowe

`mysql_db_query` - Wybiera bazę danych MySQL i wykonuje na niej zapytanie

`mysql_drop_db` - Usuwa bazę MySQL

`mysql_errno` - Zwraca liczbową wartość kodu błędu z poprzedniej operacji MySQL

`mysql_error` - Zwraca tekst komunikatu błędu z ostatniej operacji MySQL

`mysql_escape_string` - Dodaje znaki unikowe na użytek `mysql_query`

`mysql_fetch_array` - Zapisuje wiersz wyniku w tablicy asocjacyjnej, numerycznej lub w obu

`mysql_fetch_assoc` - Zapisuje wiersz wyniku w tablicy asocjacyjnej

`mysql_fetch_field` - Pobiera z wyniku informacje o kolumnie i zwraca jako obiekt

`mysql_fetch_lengths` - Pobiera długość każdego pola w wierszu wyniku

`mysql_fetch_object` - Zapisuje wiersz wyniku jako obiekt

`mysql_fetch_row` - Zapisuje wiersz wyniku w tablicy wyliczeniowej

`mysql_field_flags` - Pobiera flagi dla danego pola w wyniku

mysql_field_len - Zwraca długość danego pola

mysql_field_name - Podaje nazwę danego pola w wyniku

mysql_field_seek - Przesuwa wskaźnik wyniku do wybranego pola

mysql_field_table - Pobiera nazwę tabeli w której znajduje się dane pole

mysql_field_type - Pobiera typ podanego pola

mysql_free_result - Zwalnia pamięć zajmowaną przez wynik

mysql_get_client_info - Pobiera informacje o kliencie MySQL

mysql_get_host_info - Pobiera informacje o hoście MySQL

mysql_get_proto_info - Pobiera informacje o protokole MySQL

mysql_get_server_info - Pobiera informacje o serwerze MySQL

mysql_info - Pobiera informacje o ostatnim zapytaniu

mysql_insert_id - Podaje numer ID wygenerowany podczas ostatniej operacji INSERT

mysql_list_dbs - Pobiera nazwy baz dostępnych na serwerze MySQL

mysql_list_fields - Pobiera informacje o polach tabeli MySQL

mysql_list_processes - Zwraca listę procesów MySQL

mysql_list_tables - Pobiera listę tabel z bazy MySQL

mysql_num_fields - Zwraca liczbę pól w wyniku

mysql_num_rows - Zwraca liczbę wierszy w wyniku

mysql_pconnect - Otwiera stałe połączenie z serwerem MySQL

mysql_ping - Sprawdza połączenie z serwerem, lub nawiązuje połączenie, jeśli go nie było

mysql_query - Wysyła zapytanie do serwera MySQL

mysql_real_escape_string - Dodaje znaki unikowe w łańcuchu znaków do użycia w instrukcji SQL

mysql_result - Zwraca dane wynikowe

mysql_select_db - Wybiera bazę MySQL

mysql_set_charset - Sets the client character set

mysql_stat - Pobiera aktualny stan systemu

mysql_tablename - Zwraca nazwę tabeli

mysql_thread_id - Zwraca id bieżącego wątku

mysql_unbuffered_query - Wysyła zapytanie do serwera MySQL nie buforując wyniku

Przykłady wdrożenia wymaganych elementów

Ograniczniki

```
$dane="SELECT id_produktu, nazwa, kategoria, popularnosc FROM  
baza_produktow ORDER BY popularnosc DESC LIMIT 5";
```

Użyte w pliku zestawienia.php

Związki

Związki zostały wdrożone w prawie wszystkich tabelach, są one wymagane do poprawnego działania systemu. Przykład:

```
CREATE TABLE logowanie(  
id_klienta int not null,  
login_klienta varchar(20) not null,  
haslo_klienta varchar(200) not null,  
pracownik boolean,  
CONSTRAINT log_pk PRIMARY KEY(id_klienta),  
CONSTRAINT log_fk FOREIGN KEY(id_klienta) REFERENCES  
klienci(id_klienta)  
);
```

Widoki

```
CREATE VIEW magazyn_widok AS SELECT * from magazyn WHERE dostawca  
LIKE "Nvidia";
```

Procedury

```
CREATE DEFINER=`root`@`localhost` PROCEDURE  
`PokazZamowieniaKlienta`(IN `idzam` INT(5)) NOT DETERMINISTIC NO SQL  
SQL SECURITY DEFINER SELECT * FROM zamowienie where  
id_klienta=idzam;
```

Wyzwalacze

```
CREATE TRIGGER `STAN` BEFORE UPDATE ON `zamowienie`  
FOR EACH ROW SET NEW.data_zmiany = CURRENT_TIMESTAMP;
```

Transakcje

```
$polaczenie->begin_transaction();  
    $dane="INSERT INTO  
dane_zamowienia(id_produktu,id_zamowienia,ilosc_zamowionych)  
VALUES(".$id.", ".$wiersz['id_zamowienia']." , ".$ilosc.");";  
    $polaczenie->query($dane);  
  
    $dane="UPDATE baza_produkow SET  
popularnosc=popularnosc+".$ilosc." WHERE id_produktu=".$id."";  
    $polaczenie->query($dane);  
  
    $dane="UPDATE magazyn SET ilosc=ilosc-".$ilosc." WHERE  
id_produktu=".$id."";  
    $polaczenie->query($dane);  
  
    $dane="SELECT ilosc from magazyn WHERE  
id_produktu=".$id."";  
    $odpowiedz=$polaczenie->query($dane);  
    $wiersz=$odpowiedz->fetch_assoc();  
    if ($wiersz['ilosc']<0){  
        $polaczenie->rollback();  
        $_SESSION['zamowiono']="Wystąpił błąd, nie mamy na stanie  
tylu sztuk.";  
        break;  
        unset($_SESSION['koszyk']);  
        header('location:main.php');  
    }  
    else {
```



```

        $_SESSION['zamowiono']="Dziękujemy za złożenie
zamówienia w naszym sklepie";
    }
}
$połączenie->commit();
unset($_SESSION['koszyk']);

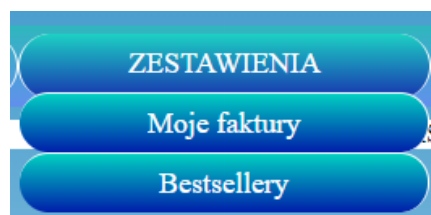
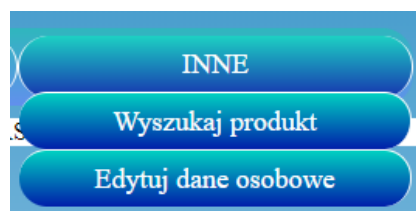
```

Interfejs użytkownika

Strona główna konta klienta



Poszczególne opcje menu klienta



Strona główna konta pracownika



Opis wdrożenia i testy

Wdrożenie zostało wykonane na platformie złożonej z systemu windows 10 oraz serwera XAMPP. Aplikacja została napisana w czystym języku PHP oraz HTML i CSS, bez użycia frameworków i bibliotek. Większość działań na bazie jest obsługiwana przez formularze html wysyłające dane do skryptów PHP metodą POST (aby utrudnić wyciek danych). Hasło służące użytkownikowi do logowania jest zapisane w bazie w formie niejawnej (MD5), logowanie jest zabezpieczone przed iniekcjami SQL użyciem funkcji htmlentities oraz sprintf, mysqli_real_escape_string. Strony dostępne tylko dla pracowników są zabezpieczone przez dostępem zwykłych użytkowników poprzez sprawdzenie uprawnień w tabeli sesji.

Możliwości działania aplikacji

W celu złożenia zamówienia należy wybrać przedmiot, wpisać liczbę i kliknąć przycisk „dodaj do koszyka” Po poprawnym dodaniu przedmiotu zostaniemy skierowani z powrotem na ekran wyboru towaru gdzie można dodać do koszyka pozostałe interesujące nas przedmioty.

NAZWA	CENA(ZŁ/SZT)	ILOŚĆ	SZT	Dodaj do koszyka
1 - LG G21	412.99	25	<input type="text" value="5"/>	Dodaj do koszyka
2 - GOODRAM BR12 DDR3	79.3	40	<input type="text"/>	Dodaj do koszyka
3 - INTEL i3 4500K	315.8	10	<input type="text"/>	Dodaj do koszyka
4 - HUB USB X3	25.69	20	<input type="text"/>	Dodaj do koszyka
5 - Karta sieciowa LAN	65.4	50	<input type="text"/>	Dodaj do koszyka
6 - EZIO Z450 27"	685.42	27	<input type="text"/>	Dodaj do koszyka
7 - GeForce GTX1080	2540	3	<input type="text"/>	Dodaj do koszyka
8 - AMD FX434	365.99	42	<input type="text"/>	Dodaj do koszyka
9 - Obudowa Aerocool Q34	99.85	30	<input type="text"/>	Dodaj do koszyka
10 - Klawiatura Corsair Strafe	100.23	18	<input type="text"/>	Dodaj do koszyka

ILOŚĆ	SZT	Dodaj do koszyka
25	<input type="text" value="26"/>	Dodaj do koszyka
40	<div>! Wartość nie może być większa niż 25.</div> <input type="text"/>	Dodaj do koszyka

Skrypt pilnuje abyśmy nie dodali do koszyka większej liczby niż aktualnie dostępna.

NAZWA	CENA(ZŁ/SZT)	ILOŚĆ	RAZEM	Usuń
LG G21	412.99	25	10324.75	Usuń
HUB USB X3	25.69	10	256.9	Usuń
EZIO Z450 27"	685.42	2	1370.84	Usuń
Łączna wartość produktów w koszyku: 11952.49zł				
<input type="button" value="złóż zamówienie"/>				
<input type="button" value="Kasuj Koszyk"/>				

Po dodaniu wszystkich towarów można przejść do koszyka w którym widzimy podsumowanie, mamy możliwość usunięcia przedmiotów oraz wysłanie zamówienia. Po wysłaniu zamówienia dane wędrują

do kilku tablic przechowujących dane, od tego momentu możemy przeglądać zamówienia i generować dokumenty.

1	1	2020-01-14 12:35:56	realizacja	2064.95	Szczegóły
2	1	2020-01-14 18:58:05	zamowione	11952.5	Szczegóły

Mamy możliwość przeglądania top5 najczęściej kupowanych produktów.

TOP 5 NAJCZĘŚCIEJ KUPOWANYCH PRODUKTÓW			
ID	NAZWA	KATEGORIA	SPRZEDANYCH
1	LG G21	monitory	30
4	HUB USB X3	akcesoria	10
6	EZIO Z450 27"	monitory	2
10	Klawiatura Corsair Strafe	akcesoria	0
3	INTEL i3 4500K	hardware	0

W tym momencie następuje przejście na konto pracownika.

Z tej pozycji mamy możliwość zarządzania zamówieniami.

Nr Zam.	ID Klienta	Data zamówienia	Aktualny stan	Nowy Stan	Zmień	Data zmiany
1	1	2020-01-16 12:35:56	realizacja	Zamówione ▾	Zmień stan	2020-01-16 12:35:56
2	1	2020-01-16 18:58:05	zamowione	Zamówione ▾	Zmień stan	0000-00-00 00:00:00

Nr Zam.	ID Klienta	Data zamówienia	Aktualny stan	Nowy Stan	Zmień	Data zmiany
1	1	2020-01-16 12:35:56	realizacja	Zamówione ▾	Zmień stan	2020-01-16 12:35:56
2	1	2020-01-16 19:02:18	realizacja	Zamówione ▾	Zmień stan	2020-01-16 19:02:18

Pracownik ma możliwość wyświetlenia faktury, która po zmianie statusu na realizację jest generowana i dostępna zarówno dla klienta i pracownika.

Zam.	ID Kl.	Data zamówienia	Status	Kwota	***
1	1	2020-01-16 12:35:56	realizacja	2064.95	Wyświetl Fakturę
2	1	2020-01-16 19:02:18	realizacja	11952.5	Wyświetl Fakturę

Dane kupującego: Jurek Owsiak Ul: Wrzeciono Poczt : 23851 Warszawa Tel: 446112113 NIP: 753-003-22-21	Dane Sprzedającego: Hurtownia WWW Sp. Cywilna Ul: Abstynencji 16 Poczt: 33300 Nowy Sącz Tel: 517312374 NIP: 7341278556
--	--

LP	NAZWA	CENA	ILOŚĆ	RAZEM
1	LG G21	412.99zł	25	10324.75zł
4	HUB USB X3	25.69zł	10	256.9zł
6	EZIO Z450 27"	685.42zł	2	1370.84zł
				Wartość netto: 9717.47zł Wartość brutto: 11952.49zł

Faktury są przygotowane do wydruku.

Wyszukiwanie produktów

Wpisz wyszukiwaną frazę:

Szukaj według:

- ☐ Kategoria
☐ Nazwa
☐ Numer katalogowy

Szukaj

Istnieje również możliwość wyszukiwania towarów po numerach katalogowych, nazwach i kategoriach.

Wnioski

Zaprojektowanie i stworzenie aplikacji która jednocześnie będzie spełniała wszystkie wymagania klienta, będąc przy tym szybką, bezpieczną i wydajną jest bardzo trudne ale nie niemożliwe. Na rynku jest dostępne bardzo dużo rozwiązań i półproduktów które mogą znaczenie ograniczyć problemy związane z zabezpieczeniami i wydajnością ale przy pewnej dozie chęci i samozaparcia można napisać własną aplikację nawet bez użycia frameworków i bibliotek komercyjnych. Rynek E-commerce jest prawdopodobnie najbardziej przesyconym rynkiem rozwiązań informatycznych gdzie każdy detalista i

hurtownik znajdzie oprogramowanie praktycznie skrojone na jego miarę. Warto jednak podjąć wyzwanie i spróbować napisać aplikację tego typu aby zobaczyć z jaką ilością problemów trzeba się zmierzyć aby stworzyć coś, co przynajmniej będzie spełniać najbardziej podstawowe wymagania.

Literatura

PHP i MySQL dla każdego. - Wydanie III Marcin Lis wyd. Helion

<https://pl.wikipedia.org/wiki/InnoDB>

[https://pl.wikipedia.org/wiki/Postać_normalna_\(bazy_danych\)](https://pl.wikipedia.org/wiki/Postać_normalna_(bazy_danych))