



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA



**Katedra
Informatyki i Automatyki**
Politechnika Rzeszowska

Bazy danych

Dokumentacja Projektu

pt.: „Sklep muzyczny”

Data wykonania: 21.01.2020

Grupa: L03
Miłosz Stączek

Spis treści

Wstęp teoretyczny	3
1. Cel i założenia projektu	4
2. Wstępny opis projektu	4
3. Wykorzystane technologie	5
4. Diagram związków encji bazy danych	6
5. Opis tabel bazy danych i ich funkcji	7
6. Funkcje bazy danych realizowane przez SQL	8

Wstęp teoretyczny

Baza danych jest to zbiór uporządkowanych informacji przechowywanych w systemie komputerowym. Bazą danych steruje się za pomocą Systemu zarządzania bazami danych (DBMS). DBMS stanowi interfejs między bazą danych a jego użytkownikami. Dane zawarte w bazie danych, system DBMS oraz powiązane z nimi aplikacje tworzą system bazodanowy. Dane umieszczane są w wierszach i kolumnach tabel, co umożliwia łatwiejsze zarządzanie danymi, ich wyszukiwanie oraz modyfikowanie. Większość baz danych wykorzystuje język strukturalny SQL (Strukturę Query Language) do tworzenia zapytań do bazy. Rozróżnia się kilka typów baz danych m.in. relacyjne, obiektowe, rozproszone.

1. Cel i założenia projektu

Celem projektu było przedstawienie zakresu i tematyki projektu, zagadnień związanych z tematem, określenie funkcji bazy danych oraz ich priorytetu, wybór narzędzi i technologii do realizacji projektu. Prezentacja opisu tabel bazy danych oraz prezentacja Diagramu związków encji. Implementacja bazy danych, prezentacja jej funkcjonalności oraz funkcji realizowanych przez SQL, dostarczenie repozytorium wraz z dokumentacją projektu.

2. Wstępny opis projektu

Tematem projektu będzie system obsługujący stacjonarny sklep muzyczny sprzedający sprzęt oraz albumy muzyczne. System będzie się zajmował przechowywaniem danych personalnych klientów, pracowników jak również ich adresy. System będzie pozwalał na dodawanie, usuwanie i modyfikację danych osobistych klientów i pracowników. Możliwe będzie przypisywanie klientom zakupionych przez nich towarów, klienci mogą podać swoje dane personalne jednak nie muszą tego robić, aby zakupić dany produkt. System będzie przede wszystkim przypisywał odpowiednie informacje dotyczące tego jaki produkt został kupiony i w jakiej ilości, ile danego towaru pozostało dostępne na magazynie oraz jaki pracownik sprzedał dany towar. Tabele, które planuję umieścić w projekcie: klienci, pracownicy, produkty, adresy, sprzedaz. Będzie to relacyjna baza danych. Będę korzystał z narzędzia phpMyAdmin do stworzenia i obsługi systemu zarządzania bazami danych MySQL.

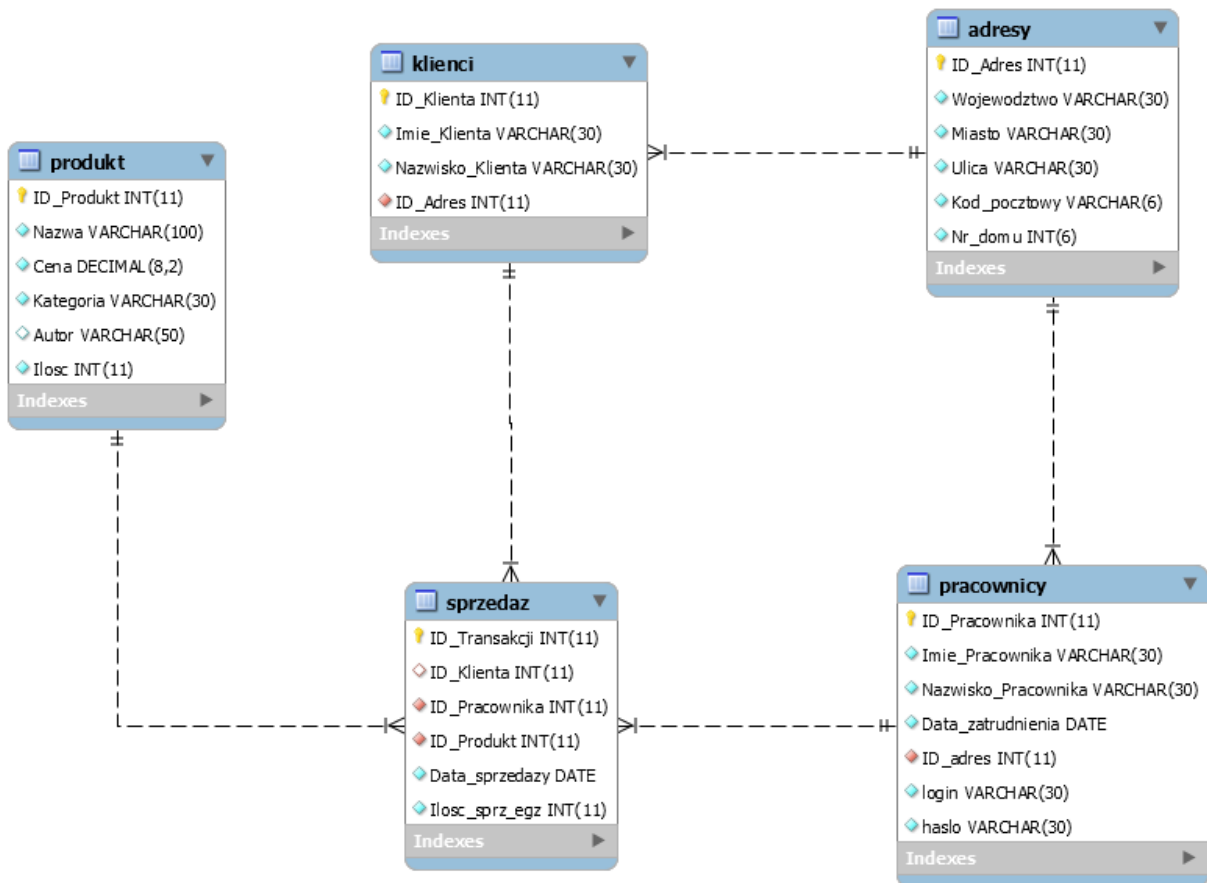
3. Wykorzystane technologie

W projekcie wykorzystano darmowy pakiet XAMPP do postawienia serwera oraz obsługi phpMyAdmin. Jest to wieloplatformowy, zintegrowany pakiet, składający się głównie z serwera Apache, bazy danych MySQL i interpreterów dla skryptów napisanych w PHP i Perlu. Pakiet jest wydawany na licencji GNU jako darmowy serwer WWW do obsługi dynamicznych stron. Obecnie XAMPP jest dostępny na cztery platformy: Microsoft Windows, Linux, Sun Solaris oraz OS X.

phpMyAdmin jest to narzędzie pozwalające w łatwy sposób zarządzać bazą danych, napisane w języku PHP wydawane na licencji GNU. Prace nad phpMyAdmin rozpoczął Tobias Ratschiller w 1998 roku, autor jednak zawiesił prace nad programem i projekt został przejęty przez trzech innych programistów i we wrześniu 2000 roku doczekał się wydania kolejnej wersji. Od tego czasu program jest nieprzerwanie rozwijany.

MySQL jest to system zarządzania relacyjnymi bazami danych, rozwijany obecnie przez firmę Oracle. MySQL sprawdza się zarówno dla małych jak i dużych projektów, jest dobrze zoptymalizowany pod kątem szybkości oraz jest łatwy w użytkowaniu, używa standardowego języka SQL. Dane przechowywane są w kolumnach i wierszach tabel połączonych relacjami.

4. Diagram związków encji bazy danych



Rysunek 1 Diagram Związków encji

5. Opis tabel bazy danych i ich funkcji

1. Klienci - Tabela "Klienci" odpowiedzialna jest za przechowywanie informacji o klientach, którzy byli chętni do podania swoich danych osobowych. Imię, Nazwisko i Adres jako klucz obcy. Każdy klient, który podał dane musi mieć przypisany adres.
2. Pracownicy - Tabela "Pracownicy" przechowuje informacje o pracownikach pracujących w danym sklepie. Imię, Nazwisko, Adres jako klucz obcy oraz datę zatrudnienia. Każdy pracownik musi mieć przypisany adres.
3. Adresy - Tabela "Adresy" przechowuje adresy klientów i pracowników. Adres przypisywany jest do Klienta lub Pracownika. Zawiera pola województwo, miasto, ulica, kod_pocztowy, nr_domu.
4. Sprzedaz - Tabela "Sprzedaz" zawiera dane łączące Pracownika, Produkt, który został sprzedany, ewentualnego klienta (jeśli ten zgodził się podać swoje dane), datę sprzedaży oraz ilość sprzedanych egzemplarzy danego produktu. Tabela ma za zadanie gromadzić informacje o tym, ile jakich produktów zostało sprzedanych o jakim czasie.
5. Produkt - Tabela "Produkt" zawiera informacje o produkcie, jego nazwę, kategorię, do której należy produkt, ewentualnego autora danego produktu (jeśli ten istnieje np. autor książki), cenę produktu oraz ilość dostępnych egzemplarzy danego produktu.

6. Funkcje bazy danych realizowane przez SQL

a) Zapytanie wyszukiujące gitary

```
SELECT * FROM produkt  
WHERE produkt.Nazwa LIKE '%gitara%';
```

Zapytanie wyświetli informacje o wszystkich rekordach z tabeli produkt które w nazwie zawierają słowo gitara.

ID_Produkt	Nazwa	Cena	Kategoria	Autor	Ilosc
103	Gitara Yamaha RGX 121 Z BL	1399.90	Instrument	NULL	3

Rysunek 2 Wszystkie pola z gitarą w nazwie

b) Zapytanie modyfikujące nazwę konkretnego produktu

```
UPDATE produkt SET nazwa ='Whitestar'  
WHERE produkt.ID_Produkt=101;
```

Zapytanie, które modyfikuje podaną wartość pola (w tym przypadku pole „nazwa”), dla rekordu, w którym spełniony jest podany warunek (w tym przypadku rekord z ID =1).

c) Zapytanie usuwające konkretny rekord

```
DELETE FROM produkt WHERE ID_Produkt='103';
```

Zapytanie usuwa podany rekord z podanej tabeli. W tym przypadku usuwany jest rekord z tabeli „produkt” z ID równym 3.



d) Zapytania dodające klienta i pracownika

```
INSERT INTO adresy(ID_Adres,Wojewodztwo,Miasto,Ulica,Kod_pocztowy,Nr_domu)
VALUES (13,'Podkarpackie','Jaslo','Robotnicza','32-144','18');

INSERT INTO klienci (ID_Klienta,Imie_Klienta,Nazwisko_Klienta,ID_Adres)
VALUES (13,'Przemyslaw','Pioro', 13);
```

Pierwsze zapytanie dodaje nowy rekord do tabeli adresy uzupełniając podane pola (ID_Adres, Wojewodztwo, Miasto, Ulica, Kod_pocztowy, Nr_domu) podanymi wartościami (13, Podkarpackie, Jaslo, Robotnicza, 32-144, 18). Następne zapytanie dodaje nowy rekord do tabeli klienci uzupełniając pola w taki sam sposób jak poprzednie.

```
INSERT INTO adresy (ID_Adres,Wojewodztwo,Miasto,Ulica,Kod_pocztowy,Nr_domu)
VALUES (85,'Podkarpackie','Przemysl','Parkowa','36-654','54');

INSERT INTO pracownicy (ID_Pracownika,Imie_Pracownika,Nazwisko_pracownika,
Data_zatrudnienia, id_adres, login, haslo)
VALUES (85,'Kazimierz', 'Zajac', '2020/01/01',85,'login5', 'haslo5');
```

Zapytania dodają informacje dotyczące nowego pracownika, takie jak jego adres oraz jego dane personalne.



e) Zapytanie wypisujące adresy klientów i pracowników

```
SELECT klienci.Imie_Klienta, klienci.Nazwisko_Klienta, adresy.Miasto,  
adresy.Ulica, adresy.Nr_domu  
FROM klienci  
INNER JOIN adresy ON klienci.ID_Adres = adresy.ID_Adres  
ORDER BY klienci.Nazwisko_Klienta asc;
```

Zapytanie, które wypisuje z bazy danych imię i nazwisko klienta z tabeli „klienci” a także informacje na temat adresu zamieszkania (miasto, ulica, nr domu) z tabeli „adresy”.
Zapytanie sortuje alfabetycznie po nazwisku klienta.

Imie_Klienta	Nazwisko_Klienta	▲ 1	Miasto	Ulica	Nr_domu
Oliwia	Marciniak		Debica	Główna	514
Tomasz	Nowak		Krakow	Kwiatowa	61

Rysunek 3 Spis klientów wraz z ich miejscem zamieszkania

```
SELECT pracownicy.Imie_Pracownika, pracownicy.Nazwisko_Pracownika,  
adresy.Miasto, adresy.Ulica, adresy.Nr_domu  
FROM pracownicy  
INNER JOIN adresy ON pracownicy.ID_adres = adresy.ID_Adres  
ORDER BY pracownicy.Nazwisko_Pracownika asc;
```

Zapytanie wypisująca informacje o pracownikach oraz ich adresy.

Imie_Pracownika	Nazwisko_Pracownika	▲ 1	Miasto	Ulica	Nr_domu
Sebastian	Drozd		Sanok	Słowackiego	54
Andrzej	Kowalczyk		Krosno	Słoneczna	21
Jan	Kowalski		Rzeszow	Mickiewicza	34
Paweł	Orkiszewski		Poznań	Kwiatowa	78

Rysunek 4 Spis pracowników wraz z ich adresem zamieszkania



f) Zapytania wyświetlające raport ze sprzedaży

```
SELECT

    sprzedaz.ID_Transakcji AS ID, produkt.Nazwa,
    CONCAT (klienci.Imie_Klienta , ' ', klienci.Nazwisko_Klienta) AS
    'Klient',
    CONCAT (pracownicy.Imie_Pracownika , ' ',
    pracownicy.Nazwisko_Pracownika) AS 'Pracownik',
    sprzedaz.Data_sprzedazy,
    sprzedaz.Ilosc_sprz_egz AS Ilosc,
    CASE WHEN klienci.ID_Klienta IS NOT NULL
        THEN ((produkt.Cena * sprzedaz.Ilosc_sprz_egz) * 0.9)
        ELSE (produkt.Cena * sprzedaz.Ilosc_sprz_egz)
    END AS Wartosc
FROM sprzedaz

    INNER JOIN produkt ON sprzedaz.ID_Produkt = produkt.ID_Produkt
    LEFT JOIN klienci ON sprzedaz.ID_Klienta = klienci.ID_Klienta
    INNER JOIN pracownicy ON sprzedaz.ID_Pracownika =
    pracownicy.ID_Pracownika

ORDER BY ID_Transakcji ASC;
```

Zapytania wyświetlają wszystkie przeprowadzone transakcje. Wyświetlana jest nazwa zakupionego produktu, klient, który zakupił dany towar (o ile ten zgodził się podać swoje dane osobowe i figuruje w systemie). Pracownik, który przeprowadził daną transakcję, datę sprzedaży, ilość sztuk sprzedanego towaru oraz wartość końcową danej transakcji. Zapytania liczą końcową wartość mnożąc ilość zakupionych sztuk przez cenę jednostkową danego towaru. Jeśli transakcji dokonał klient, który figuruje w systemie zostanie mu naliczony rabat w wysokości 10% od ceny końcowej. Zapytanie sortuje po ID transakcji.

ID	Nazwa	Klient	Pracownik	Data_sprzedazy	Ilosc	Wartosc
1001	Blackstar	NULL	Jan Kowalski	2020-01-01	2	91.800
1002	Blackstar	Oliwia Marciniak	Andrzej Kowalczyk	2020-01-01	2	82.620
1003	Fortepian YAMAHA CLP-665GP	Tomasz Nowak	Sebastian Drozd	2020-01-02	1	13077.900
1004	A Night at the Opera	NULL	Sebastian Drozd	2020-01-03	3	97.500
1005	The Rise and Fall of Ziggy Stardust and the Spider...	NULL	Jan Kowalski	2020-01-05	1	24.900
1006	Invisible Touch	NULL	Paweł Orkiszewski	2020-01-06	1	29.900
1007	Skrzypce Ever Play EV613	Tomasz Nowak	Paweł Orkiszewski	2020-01-06	2	756.000

Rysunek 5 Raport ze sprzedaży

g) Zapytanie zliczające wartości zamówień dla poszczególnych klientów

```
SELECT
    COUNT(sprzedaz.ID_Transakcji) AS Ilosc_Transakcji,
    CONCAT (klienci.Imie_Klienta, ' ', klienci.Nazwisko_Klienta) AS 'Klient',
    SUM(produkt.Cena * sprzedaz.Ilosc_sprz_egz * 0.9) AS Wartosc
FROM sprzedaz
    INNER JOIN produkt ON sprzedaz.ID_Produkt = produkt.ID_Produkt
    INNER JOIN klienci ON sprzedaz.ID_Klienta = klienci.ID_Klienta
GROUP BY klienci.ID_Klienta
ORDER BY klienci.ID_Klienta;
```

Zapytania zliczają, ile transakcji wykonał dany klient, wyświetla jego imię i nazwisko, oraz wartość całkowitą jego dotychczasowych transakcji. Transakcje anonimowe nie zostają uwzględnione. Zapytania sortują spis po ID klienta.

Ilosc_Transakcji	Klient	Wartosc
2	Tomasz Nowak	13833.900
1	Oliwia Marciniak	82.620

Rysunek 6 Ilość oraz wartość całkowita transakcji klientów

h) Zapytania pozwalające sprzedać dany produkt.

```
SELECT ID_Produkt AS ID, Nazwa, Cena, Ilosc FROM produkt WHERE Nazwa LIKE '%ziggy%';
```

Powyższe zapytanie pokazuje produkt, który ma zostać sprzedany. Wyświetlana jest jego ID, nazwa, cena jednostkowa oraz ilość dostępnych sztuk danego produktu. Służy ono do sprawdzenia, ile towary danego typu jest dostępnego w danym momencie w sklepie.

ID	Nazwa	Cena	Ilosc
102	The Rise and Fall of Ziggy Stardust and the Spider...	24.90	13

Rysunek 7 Informacje na temat konkretnego produktu

```
INSERT INTO
sprzedaz(ID_Klienta,ID_Pracownika,ID_Produkt,Data_sprzedazy,Ilosc_sprz_egz)
VALUES
(NULL, 82, 102, CURRENT_DATE, 1);

UPDATE produkt, sprzedaz
SET produkt.Ilosc = (produkt.Ilosc - sprzedaz.Ilosc_sprz_egz)
WHERE sprzedaz.ID_Transakcji=LAST_INSERT_ID()
AND produkt.ID_Produkt = sprzedaz.ID_Produkt;
```

Zapytanie INSERT INTO pozwala dodać nowy rekord do tabeli „sprzedaz”. Dodany rekord zawiera id transakcji (dodawany automatycznie poprzez funkcję auto_increment), id klienta (o ile ten zgodził się podać swoje dane osobowe – w tym przypadku NULL – sprzedaż anonimowa) id pracownika, id produktu, obecną datę oraz ilość kupowanych podczas tej transakcji egzemplarzy.

Zapytanie UPDATE modyfikuje wartość pola „ilosc” w tabeli „produkt” odejmując ilość sprzedanych w danej transakcji od ilości całkowitej dostępnej w danym momencie w sklepie.

ID	Nazwa	Klient	Pracownik	Data_sprzedazy	Ilosc	Wartosc
1008	The Rise and Fall of Ziggy Stardust and the Spider...	NULL	Andrzej Kowalczyk	2020-01-20	1	24.900

Rysunek 8 Nowo dodana sprzedaż

ID	Nazwa	Cena	Ilosc
102	The Rise and Fall of Ziggy Stardust and the Spider...	24.90	12

Rysunek 9 Zmodyfikowane pole "ilosc" po wykonaniu zapytań