



Politechnika Rzeszowska
Wydział Elektrotechniki i Informatyki

Katedra Informatyki i Automatyki

Bazy Danych

PREZENTACJA SQL

Temat: „Implementacja bazy danych dla szkoły ogrodniczej”

Wykonał:
MIZERA DAWID
L02 161885
III EF-ZI

Rzeszów 2020

1. Prezentacja przykładowych zapytań.

1.1 Tworzenie tabel.

1.1.1 Tabela galeria.

```
CREATE TABLE galeria(  
  id_zdjecie INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  id_produkt INT,  
  url VARCHAR (3000),  
  data_dodania DATETIME  
);
```

1.1.2 Tabela produkt.

```
CREATE TABLE produkt(  
  id_produkt INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  id_magazyn INT,  
  nazwa VARCHAR (45),  
  typ VARCHAR(45),  
  cena_netto DECIMAL(10,2),  
  cena_brutto DECIMAL(10,2),  
  opis TEXT  
);
```

1.1.3 Tabela magazyn.

```
CREATE TABLE magazyn(  
  id_magazyn INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  kod_produkt VARCHAR(45),  
  ilość_pierwszy_gatunek INT,  
  ilość_drugi_gatunek INT,  
  lokalizacja VARCHAR(45)  
);
```

1.1.4 Tabela pracownik.

```
CREATE TABLE pracownik(  
  id_pracownik INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  imie VARCHAR(45) NOT NULL,  
  nazwisko VARCHAR(45) NOT NULL,  
  id_adres INT,  
  id_kontakt INT ,  
  login VARCHAR(45) NOT NULL,  
  haslo VARCHAR(45) NOT NULL,  
  uprawnienia ENUM('0','1','2','3') NOT NULL,  
  konto_aktywne BOOLEAN NOT NULL,  
  data_zatrudnienia DATETIME NOT NULL,  
  data_zwolnienia DATETIME  
);
```

1.1.5 Tabela zamowienie_produk.

```
CREATE TABLE zamowienie_produk(  
  id_zamowienie_produk INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  id_zamowienie INT,  
  id_produk INT,  
  ilosc INT  
);
```

1.1.6 Tabela adres.

```
CREATE TABLE adres(  
  id_adres INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  miejscowosc VARCHAR(45) NOT NULL,  
  powiat VARCHAR(45) NOT NULL,  
  wojewodztwo VARCHAR(45) NOT NULL,  
  kraj VARCHAR(45) NOT NULL,  
  kod_pocztowy VARCHAR(10) NOT NULL,  
  ulica VARCHAR(45),  
  nr_domu VARCHAR(10),  
  nr_mieszkania VARCHAR(10)  
);
```

1.1.7 Tabela zamowienie.

```
CREATE TABLE zamowienie(  
  id_zamowienie INT PRIMARY KEY,  
  id_klient INT,  
  data_zlozenia_zamowienia DATETIME,  
  czy_przyjeto_zamowienie BOOLEAN,  
  data_przyjecia_zamowienia DATETIME,  
  forma_zaplaty VARCHAR(45),  
  metoda_wysylki VARCHAR(45),  
  status_zamowienia VARCHAR(45),  
  data_realizacji_zamowienia DATETIME,  
  status_zaplaty VARCHAR(45)  
);
```

1.1.8 Tabela kontakt.

```
CREATE TABLE kontakt(  
  id_kontakt INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  nr_telefonu_1 VARCHAR(45) NOT NULL,  
  nr_telefonu_2 VARCHAR(45),  
  fax VARCHAR(45),  
  email VARCHAR(45) NOT NULL  
);
```

1.1.9 Tabela klient.

```
CREATE TABLE klient(  
id_klient INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
id_adres INT ,  
id_kontakt INT,  
nazwisko VARCHAR(45) NOT NULL,  
imie VARCHAR(45) NOT NULL,  
login VARCHAR(45) NOT NULL,  
haslo VARCHAR(45) NOT NULL  
);
```

1.2 Wstawianie kluczy obcych.

```
ALTER TABLE `galeria` ADD FOREIGN KEY (`id_produkt`) REFERENCES `produkt`(`id_produkt`);  
  
ALTER TABLE `produkt` ADD FOREIGN KEY (`id_magazyn`) REFERENCES `magazyn`(`id_magazyn`);  
  
ALTER TABLE `zamowienie_produkt` ADD FOREIGN KEY (`id_zamowienie`) REFERENCES `zamowienie`(`id_zamowienie`);  
  
ALTER TABLE `zamowienie_produkt` ADD FOREIGN KEY (`id_produkt`) REFERENCES `produkt`(`id_produkt`);  
  
ALTER TABLE `zamowienie` ADD FOREIGN KEY (`id_klient`) REFERENCES `klient`(`id_klient`);  
  
ALTER TABLE `klient` ADD FOREIGN KEY (`id_adres`) REFERENCES `adres`(`id_adres`);  
  
ALTER TABLE `klient` ADD FOREIGN KEY (`id_kontakt`) REFERENCES `kontakt`(`id_kontakt`);  
  
ALTER TABLE `pracownik` ADD FOREIGN KEY (`id_adres`) REFERENCES `adres`(`id_adres`);  
  
ALTER TABLE `pracownik` ADD FOREIGN KEY (`id_kontakt`) REFERENCES `kontakt`(`id_kontakt`);
```

1.3 Uzupełnianie tabel podanymi wartościami.

```
INSERT INTO `adres` (`miestowosc`, `powiat`, `województwo`, `kraj`, `kod_pocztowy`, `ulica`,  
`nr_domu`, `nr_mieszkania`) VALUES ('Wojnicz', 'Tarnowski', 'Małopolskie', 'Polska', '32-830', '',  
'224', '');  
  
INSERT INTO `kontakt` (`nr_telefonu_1`, `nr_telefonu_2`, `fax`, `email`) VALUES ('146791562',  
'', '835789378', 'bartek@gmail.com'), ('679365098', '', '', 'danuta@gmail.com');  
  
INSERT INTO `magazyn`  
(`kod_produkt`, `ilosc_pierwszy_gatunek`, `ilosc_drugi_gatunek`, `lokalizacja`) VALUES  
(`10`, `124`, `45`, `paleta 1`);
```

1.4 Usuwanie tabel.

```
DROP TABLE IF EXISTS `sklep`.`galeria`;  
DROP TABLE IF EXISTS `sklep`.`produkt`;  
DROP TABLE IF EXISTS `sklep`.`magazyn`;  
DROP TABLE IF EXISTS `sklep`.`pracownik`;  
DROP TABLE IF EXISTS `sklep`.`zamowienie_produkt`;
```

```
DROP TABLE IF EXISTS 'sklep'.'zamowienie';
DROP TABLE IF EXISTS 'sklep'.'adres';
DROP TABLE IF EXISTS 'sklep'.'kontakt';
DROP TABLE IF EXISTS 'sklep'.'klient'
```

1.5 Edycja rekordów w tabeli.

```
UPDATE `kontakt` SET `nr_telefonu_1` = '14567890' WHERE `kontakt`.`id_kontakt` = 7;
```

2 Prezentacja wybranych funkcji.

2.1 Funkcja zliczająca całkowitą ilość towaru jednego gatunku w magazynie.

```
DROP FUNCTION `SUMA_ODMIANY_MAGAZYN`;
CREATE DEFINER=`root`@`localhost` FUNCTION
`SUMA_ODMIANY_MAGAZYN`(`id_towaru` INT) RETURNS INT(11) NOT DETERMINISTIC
CONTAINS SQL SQL SECURITY DEFINER BEGIN
DECLARE wynik INT DEFAULT 0;
DECLARE pierwszy_gatunek INT;
DECLARE drugi_gatunek INT;
SET pierwszy_gatunek = (select ilosc_pierwszy_gatunek from magazyn where
magazyn.id_produkt_magazyn = id_towaru);
SET drugi_gatunek = (select ilosc_drugi_gatunek from magazyn where
magazyn.id_produkt_magazyn = id_towaru);
SET wynik=pierwszy_gatunek+drugi_gatunek;
RETURN wynik;
END
```

2.2 Funkcja wyszukująca najtańszy produkt w asortymencie.

```
BEGIN
DECLARE naj DECIMAL(10.2) DEFAULT 0;
SET naj =(SELECT MIN(cena_brutto) FROM produkt);
RETURN naj;
END
```

2.3 Funkcja wyszukująca najdroższy produkt w asortymencie.

```
BEGIN
DECLARE naj DECIMAL(10.2) DEFAULT 0;
SET naj =(SELECT MAX(cena_brutto) FROM produkt);
RETURN naj;
END
```

2.4 Funkcja sprawdzająca ile zostało sztuk danej odmiany w magazynie.

```
BEGIN
DECLARE wynik INT DEFAULT 0;
DECLARE wsk INT DEFAULT 0;
DECLARE ilosc INT;
```

```

SET wsk = (select id_produkt_magazyn from produkt where produkt.nazwa = nazwa);
SET ilosc = (select ilosc_pierwszy_gatunek from magazyn where magazyn.id_produkt_magazyn =
wsk);
SET wynik=ilosc;
RETURN wynik;
END

```

3. Przedstawienie wybranych procedur.

3.1 Procedura wyświetlająca zamówiony przez klienta produkt.

```

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `Wyswietl_zamowienie`(IN `Nazwisko`
VARCHAR(45))
    NO SQL
BEGIN
select zp.id_zamowienie as nr_zamowienia, p.nazwa, zp.ilosc
from zamowienie_produkt as zp
inner join produkt as p on zp.id_produkt = p.id_produkt
inner join zamowienie as z on zp.id_zamowienie = z.id_zamowienie
inner join klient as k on z.id_klient = k.id_klient
where k.nazwisko = Nazwisko;
END$$
DELIMITER ;

```

3.2 Procedura usuwająca rekord w tabeli galeria.

```

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `usun_zdjecie`(IN `id_zdjecie` INT)
    NO SQL
BEGIN
DELETE FROM galeria WHERE (id_zdjecie=id_zdjecie);
END$$
DELIMITER ;

```

3.3 Procedura edytująca wpis rekordu w tabeli adres.

```

DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `edytuj_kontakt`(IN `id_kontakt` INT, IN `nr_telefonu_1`
VARCHAR(45), IN `nr_telefonu_2` VARCHAR(45), IN `fax` VARCHAR(45), IN `email` VARCHAR(45))
    NO SQL
BEGIN

IF NOT EXISTS ( SELECT id_kontakt FROM kontakt WHERE kontakt.id_kontakt =id_kontakt)
THEN
SELECT 'Podanego ID nie ma w bazie';
ELSE

UPDATE kontakt
SET kontakt.nr_telefonu_1 = nr_telefonu_1
WHERE kontakt.id_kontakt = id_kontakt AND nr_telefonu_1 > 0;

UPDATE kontakt
SET kontakt.nr_telefonu_2= nr_telefonu_2
WHERE kontakt.id_kontakt = id_kontakt AND nr_telefonu_2>0;

```

```
UPDATE kontakt
SET kontakt.email = email
WHERE kontakt.id_kontakt = id_kontakt AND email LIKE '%@%';
```

```
UPDATE kontakt
SET kontakt.fax = fax
WHERE kontakt.id_kontakt = id_kontakt AND fax >0;
END IF;
END$$
DELIMITER ;
```

3.4 Procedura dodająca nowy rekord tabeli produkt wraz z utworzeniem nowego rekordu w tabeli magazyn.

```
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `dodaj_produkt_z_uzupelnieniem_magazynu`(IN
`kod_produkt` VARCHAR(45), IN `ilosc_pierwszy_gatunek` INT, IN `ilosc_drugi_gatunek` INT, IN `lokalizacja`
INT(45), IN `nazwa` VARCHAR(45), IN `typ` VARCHAR(45), IN `cena_netto` DECIMAL(10,2), IN `cena_brutto`
DECIMAL(10,2), IN `opis` TEXT)
NO SQL
BEGIN
INSERT INTO firma.magazyn(kod_produkt,ilosc_pierwszy_gatunek,ilosc_drugi_gatunek,lokalizacja)
VALUES(kod_produkt,ilosc_pierwszy_gatunek,ilosc_drugi_gatunek,lokalizacja);
INSERT INTO firma.produkt(id_magazyn,nazwa,typ, cena_netto,cena_brutto,opis)
VALUES(LAST_INSERT_ID(),nazwa,typ,cena_netto,cena_brutto,opis);
END$$
DELIMITER ;
```