

Na przyszłe zajęcia silnik bazy danych MSSQL Express

SQL Server Management Studio

Rodzaje baz danych:

- Relacyjna
- Obiektowo-relacyjna
- Grafowa
- Obiektowa

Podstawowym obiektem w bazie jest tabela, która składa się z:

- Wiersze
- kolumny
- klucz główny, dzięki któremu możemy łączyć dane w tabelach. Zapis PRIMARY KEY oznacza że wartość musi być unikalna, nie trzeba stosować UNIQUE
- kluczy obcych, to id z innej tabeli
- indeksów przyspieszających wyszukiwanie danych
- wartości unikatowych, wartości pojawiających się tylko raz w tabeli.

CREATE DATABASE database_name- tworzenie nowej bazy danych

CREATE TABLE [user] – tworzenie tabeli o nazwie user.

```
CREATE TABLE USER(  
    ID NUMBER PRIMARY KEY NOT NULL,  
    NAZWA VARCHAR(50) NOT NULL,  
    HASLO VARCHAR(50) NOT NULL,  
    SLOWNIK_POZYCJI_ID NUMBER NOT NULL  
)
```

Zagadnienia do samodzielnego sprawdzenia:

- szyfrowanie,
- hashowanie,
- solenie hasła,
SHA-256,
SHA-512,
SHA-1024

```
CREATE TABLE SLOWNIK_POZYCJI(  
    ID NUMBER(40) PRIMARY KEY NOT NULL,  
    POZYCJA VARCHAR(50) NOT NULL UNIQUE  
)
```

```
CREATE TABLE ADRESY(  
    ID NUMBER PRIMARY KEY NOT NULL,  
    IMIE VARCHAR(50) NOT NULL,
```

```
NAZWISKO VARCHAR(50) NOT NULL,  
ULICA VARCHAR(100) NOT NULL,  
MIASTO VARCHAR(50) NOT NULL  
KOD_POCZTOWY VARCHAR(6) NOT NULL,  
NUMER_DOMU VARCHAR(4) NOT NULL,  
NUMER_LOKALU VARCHAR(4)
```

)

DML- Data Manipulation Language

DDL- Data Definition Language

Szyfrowanie- Zamienianie wyrazów, na ciągi znaków, które są nie do odkodowania dla osoby nie znającej funkcji szyfrującej. Najprostszy szyfr to szyfr Cezara zwany też przestawnym.

Hashowanie- lub funkcja skrótu to przypisanie krótkiego losowego ciągu znków w stałym miejscu. Do tego procesu wykorzystujemy tablice mieszające (struktura danych) które pozwalają uzyskać klucz i wartość do klucza. np. SHA-256, SHA-512, SHA-1024

Solenie hasła- dodanie zaszyfrowanego ciągu stałych lub zmiennych znaków.

```
CREATE TABLE PRODUKTY(  
    ID NUMBER PRIMARY KEY NOT NULL,  
    NAZWA VARCHAR(250) NOT NULL,  
    SKU NUMBER NOT NULL,  
    CENA_JEDNOSTKOWA NUMBER(5,2) NOT NULL - - komentarz zapis NUMBER(5,2)
```

oznacza że nasza cena może mieć 5 miejsc przed przecinkiem i 2 miejsca po przecinku.

)

```
CREATE TABLE ZAMOWIENIA(  
    ID NUMBER PRIMARY KEY NOT NULL,  
    ID_ADRES NUMBER NOT NULL,  
    WARTOSC NUMBER(10,2) NOT NULL,  
    DATA DATE NOT NULL,  
    ID_TYP_DOSTAWY NUMBER NOT NULL,  
    ID_PLATNOSCI NUMBER
```

)

```
CREATE TABLE TYP_DOSTAWY(  
    ID NUMBER PRIMARY KEY NOT NULL,  
    WARTOSC VARCHAR(50) NOT NULL
```

)

```
CREATE TABLE PLATNOSCI(  
    ID NUMBER PRIMARY KEY NOT NULL,  
    TYP VARCHAR(50) NOT NULL,  
    WARTOSC NUMBER(10,2) NOT NULL,  
    POSREDNIK VARCHAR(2000) NOT NULL,  
    DATA DATE)
```

```
CREATE TABLE ZAMOW_PRODUKT(
    ID NUMBER PRIMARY KEY NOT NULL,
    ID_ZAMOWIENIE NUMBER NOT NULL,
    ID_PRODUKT NUMBER NOT NULL,
    ILOSC NUMBER NOT NULL
)
```

PRIMARY KEY powoduje unikatowość, możliwość lepszego łączenia tabel i podbijanie wartości dokładnie o jeden względem ostatniego wiersza.

FOREGIN KEY- wartość klucza obcego z innej tabeli. Dzięki temu łączymy dwie tabele. WAŻNE FK jest PK w tabeli z której pochodzi.

Typ DML

```
INSERT INTO [NAZWA_TABELI](KOLUMNA1, KOLUMNA2,...,KOLUMNAn-ta) VALUES
(WARTOSC1, WARTOSC2,..., WARTOSCn-ta)
```

WAŻNE Jeśli w tabeli mamy kolumny które są NOT NULL to muszą być wyszczególnione w insercie i muszą dostać wartości w insercie. W nawiasie values musimy wpisać tyle wartości zgodnych z typami danych ile zadeklarowaliśmy kolumn w insercie.

```
DELETE FROM [NAZWA_TABELI] WHERE [WARUNKI]
```

WAŻNE w delete potrzebujemy zdefiniować warunki.

```
UPDATE [NAZWA_TABELI]
```

```
SET KOLUMNA1=WARTOSC1, KOLUMNA2=WARTOSC2,...,KOLUMNAn-ta = WARTOSCn-ta
WHERE WARUNEK
```

Typ DDL

CREATE- tworzenie tabel, indeksów, widoków, funkcji, procedur, pakietów, triggerów

DROP – usuwanie tabel, indeksów, widoków, funkcji, procedur, pakietów, triggerów

ALTER- zmienianie tabel, indeksów, widoków, funkcji, procedur, pakietów, triggerów

Przykłady

```
CREATE TABLE[NAZWA_TABELI]
```

```
CREATE DATABASE[NAZWA_BAZY]
```

```
DROP TRIGGER[NAZWA_TRIGGERA]
```

```
DROP INDEX[NAZWA_INDEXU]
```

```
ALTER FUNCTION [NAZWA_FUNKCJI]
```

```
BEGIN
```

```
END
```

```
ALTER PROCEDURE[NAZWA_PROCEDURY]
```

```
BEGIN
```

```
END
```

1. katarzyna.zdunek87@gmail.com

Tworzenie bazy za pomocą Microsoft SQL Menegement Studio.

Tworzenie struktury bazy

Altery na bazie.

W momencie kiedy zapomnieliśmy dodać jednej kolumny stosujemy alter na bazie.

Chcemy do naszej tabeli produkty dodać typ_produkty i do tabeli adresy uzytkownik_id, gdyż zauważyliśmy że adres jest „oderwany” od użytkownika.

```
alter table PRODUKTY add typ_produkty varchar(50);
```

```
alter table ADRESY add uzytkownik_id integer
```

Inserty do bazy

Aby dodać nowe dane do poszczególnych tabel stosujemy insert

```
insert into TYP_DOSTAWY (id, WARTOSC) values (1, 'poczta');
```

```
insert into TYP_DOSTAWY (id, WARTOSC) values (2, 'kurier');
```

```
insert into TYP_DOSTAWY (id, WARTOSC) values (3, 'płatność przy odbiorze,  
kurier');
```

```
INSERT INTO TYP_DOSTAWY (id, WARTOSC) values (4, 'odbiór osobisty');
```

```
insert into PRODUKTY (id, NAZWA, sku, CENA_JEDNOSTKOWA) values (5, 'woda mineralna  
Muszynianka', 57673, 2.1),
```

```
(6, 'chleb', 1, 2.5),
```

```
(7, 'bułki', 234, 0.49);
```

```
insert into UZYTKOWNIK (id, NAZWA, haslo, SLOWNIK_POZYCJI_ID)
```

```
values (2, 'admin', 'admin', 2),
```

```
(3, 'kate', 'kate', 1),
```

```
(4, 'jankowski2', 'jankowski2', 3);
```

```
insert into ADRESY (id, IMIE, NAZWISKO, ULICA, MIASTO, KOD_POCZTOWY, NUMER_DOMU,  
NUMER_LOKALU, uzytkownik_id)
```

```
values (1, 'Jan', 'Kowski', 'Piłsudskiego', 'Kutno', 99-345, 15, 3, 4)
```

Selecty z jednej tabeli.

```
select * from PRODUKTY p
```

```
select nazwa, cena_jednostkowa from PRODUKTY p
```

```
select wartosc from TYP_DOSTAWY td
```

```
select sku, nazwa from produkty
```

```
select * from PRODUKTY p
```

```
select * from ADRESY a
```

Dlaczego tabela użytkowników nie może mieć dwa razy takiego samego użytkownika.

Ponieważ podczas logowania do systemu nie było by wiadomo które hasło jest prawidłowe.

W momencie uzupełniania tabel i robienia prostych selectów zauważyliśmy że pewne dane nam się nie zgadzają. Stąd zrobiliśmy update na tabelach w poszczególnych kolumnach

```
update PRODUKTY set typ_produkty = 'owoce'  
where id in (1, 2,3)
```

```
update produkty set typ_produkty = 'warzywa'  
where id = 4
```

```
UPDATE PRODUKTY set typ_produkty = 'napoje'  
where id = 5
```

```
update PRODUKTY set typ_produkty = 'pieczywo'  
where id in (6,7)
```

```
update adresy set KOD_POCZTOWY = '99-234'  
where id = 1
```

```
update adresy set KOD_POCZTOWY = 'ała ma kota'  
where id = 1
```

```
update UZYTKOWNIK set NAZWA = 'kate1' where id = 3
```

Zadanie do domu:

dodać dwóch użytkowników którzy są klientami

dodać dwa nowe adresy dla tych klientów. Pamiętaj o sprawdzeniu id klienta przed dodaniem mu adresu.

Uzupełnienie foregin key