

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.

FOREIGN 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
```

