

# Bazy danych

## Wykład 4\_2

**Temat: Tworzenie schematu bazy w T-SQL**

**Sławomir Świętoniowski**

slawomir-swietoniowski@wp.pl

# Plan wykładu

---

- 1. Tworzenie i modyfikowanie bazy danych.**
2. Typy danych w MS SQL Server 2008 R2.
3. Tworzenie, modyfikowanie i usuwanie tabel.

# Prosta metodologia tworzenia relacyjnych baz danych

---

1. **Analiza dziedziny:** rozpoznanie systemu i zachodzących w nim procesów, które mają być modelowane.
2. **Projekt i implementacja schematu bazy danych (2–3NF):** tabele i relacje.
3. **Projekt i implementacja więzów integralności:**
  - **wewnętrznych – deklaratywnych** (ograniczenia PK, FK, CHECK, UNIQUE, DEFAULT, reguły, wartości domyślne);
  - **dodatkowych – proceduralnych** (wyzwalacze – ang. *triggers*).
4. **Tworzenie widoków i procedur przechowywanych dla operacji CRUD**  
– instrukcje INSERT, SELECT, UPDATE, DELETE; transakcje.
5. **Projekt i implementacja indeksów** – na podstawie struktury bazy danych i sposobu jej wykorzystania.
6. **Projekt i implementacja systemu bezpieczeństwa**  
– konta, role, użytkownicy i ich uprawnienia.
7. **Wdrożenie, konfiguracja zadań automatycznych i utrzymania bazy.**

# T-SQL: Tworzenie nowej bazy

## Składnia

```
CREATE DATABASE database_name -- nazwa bazy
[ ON                                -- wskazuje, że pliki danych są podawane jawnie
[ < filespec > [ ,...n ] ]          -- pliki danych w grupie "Primary"
[ , < filegroup > [ ,...n ] ]      -- grupy plików; n - wskazuje, że może być ich wiele
]
[ LOG ON { < filespec > [ ,...n ] } ] -- definicja plików dziennika transakcji
[ COLLATE collation_name ] -- porządek sortowania i porównywania znaków
[ FOR LOAD | FOR ATTACH ]          -- opcja dbo use only | przyłączenia bazy

< filespec > ::=
[ PRIMARY ]                        -- definicja grupy plików "Primary"
( [ NAME = logical_file_name , ]   -- logiczna nazwa pliku danych
  FILENAME = 'os_file_name'        -- fizyczna nazwa pliku wraz ze ścieżką
  [ , SIZE = size ]                -- początkowy rozmiar pliku
  [ , MAXSIZE = { max_size | UNLIMITED } ] -- rozmiar maks. | nieograniczony
  [ , FILEGROWTH = growth_increment ] ) [ ,...n ] -- przyrost rozmiaru pliku

< filegroup > ::=
-- definicja grupy plików

FILEGROUP filegroup_name < filespec > [ ,...n ] -- specyfikacja nazw grupy i plików
```

# T-SQL: Tworzenie nowej bazy

## Przykłady (1)

- Proste tworzenie bazy [Uczelnia\_Wroclaw] z domyślnymi parametrami
- (m.in. ścieżki przechowywania plików, ich rozmiar początkowy i przyrost).

**CREATE DATABASE Uczelnia\_Wroclaw**

**GO** -- Komenda "GO" rozpoznawana przez m.in. Query Analyzer, sygnalizująca  
-- koniec ciągu instrukcji SQL, które powinny być przesłane do serwera.

- Usunięcie bazy [Uczelnia\_Wroclaw].
- Aby móc wykonać to polecenie, usuwana baza nie może być używana.
- (Klient nie może mieć połączenia w kontekście usuwanej bazy,
- lecz należy uruchomić najpierw np. USE master).

-- Zmiana bieżącej bazy na [master].

**USE master**

**GO**

- Usunięcie bazy.

**DROP DATABASE Uczelnia\_Wroclaw**

**GO**

# T-SQL: Tworzenie nowej bazy

## Przykłady (2)

-- Utworzenie bazy z podaniem parametrów.

**CREATE DATABASE Uczelnia\_Wroclaw**  
**ON PRIMARY**

**( NAME = Uczelnia\_Wroclaw\_dat,**

**FILENAME = 'd:\Uczelnia\_Wroclaw.mdf',** -- Lokalizacja głównego pliku bazy.

**SIZE = 20MB,** -- Rozmiar początkowy.

-- Ponieważ nie ma podanego parametru MAXSIZE, plik może rosnąć bez ograniczeń.

**FILEGROWTH = 15% ),** -- Procentowy rozrost bazy.

**LOG ON**

**( NAME = Uczelnia\_Wroclaw\_log,**

**FILENAME = 'd:\Uczelnia\_Wroclaw.ldf',** -- Lokalizacja pliku dziennika transakcji.

**SIZE = 10MB,**

**FILEGROWTH = 10MB )**

-- Porządek sortowania i porównywania znaków: polski, strona kodowa CP1250,

-- bez rozróżniania małych i wielkich liter (CI - case insensitive), z rozróżnianiem

-- znaków narodowych przy sortowaniu, np. A,Ą,B,C,Ć,... (AS - accent sensitive).

**COLLATE SQL\_Polish\_CP1250\_CI\_AS**

# Właściwości bazy danych MS SQL

---

## Wybrane właściwości:

- **Auto update statistics** – automatyczna aktualizacja statystyk (np. do optymalizacji indeksów);
- **Auto close** – baza danych jest zamykana po wyjściu z niej ostatniego użytkownika;
- **Auto shrink** – pliki bazy automatycznie zmniejszają swoje rozmiary przy kasowaniu danych;
- **Torn page detection** – SQL Server wykrywa niekompletne operacje wejścia – wyjścia.

## Możemy je sprawdzać za pomocą:

- **Enterprise Manager** – Database Properties – Options;
- funkcji „**DATABASEPROPERTYEX**”;
- procedur systemowych „**sp\_dboption**” i „**sp\_helpdb**”.

# T-SQL: Modyfikowanie bazy

## Składnia

```
ALTER DATABASE database           -- nazwa bazy danych
{ ADD FILE < filespec > [ ,...n ]   -- dodawanie nowego pliku danych
[ TO FILEGROUP filegroup_name ]    -- grupa, do której mają być dodane pliki
| ADD LOG FILE < filespec > [ ,...n ] -- dodawanie nowego pliku dziennika
| REMOVE FILE logical_file_name    -- usuwanie pliku poprzez nazwę logiczną
| ADD FILEGROUP filegroup_name     -- dodawanie grupy plików
| REMOVE FILEGROUP filegroup_name  -- usuwanie grupy plików
| MODIFY FILE < filespec >         -- modyfikowanie pliku
| MODIFY NAME = new_dbname         -- zmiana nazwy bazy danych
| MODIFY FILEGROUP filegroup_name  -- zmiana grupy plików
{filegroup_property | NAME = new_filegroup_name }
| SET < optionspec > [ ,...n ] [ WITH < termination > ] -- ustawianie opcji bazy
| COLLATE < collation_name >      -- ustawianie porządku sortowania
}
```



# T-SQL: Modyfikowanie bazy

## Przykłady (1)

---

- **Zmiana rozmiaru pliku bazy:**

```
ALTER DATABASE Uczelnia_Wroclaw  
MODIFY FILE  
    ( NAME = Uczelnia_Wroclaw_dat,  
      SIZE = 30MB)  
GO
```

- **Dodawanie nowego pliku do bazy:**

```
ALTER DATABASE Uczelnia_Wroclaw  
ADD FILE  
    (NAME = Uczelnia_Student_dat,  
     FILENAME = 'c:\data\Uczelnia_Student.ndf',  
     SIZE = 50MB,  
     MAXSIZE = 100MB,  
     FILEGROWTH = 5MB)  
GO
```

# T-SQL: Modyfikowanie bazy

## Przykłady (2)

---

- Ustawianie właściwości bazy danych:

-- Wyłączenie automatycznego, okresowego zmniejszania plików bazy.

```
ALTER DATABASE Uczelnia_Wroclaw
```

```
SET AUTO_SHRINK OFF
```

```
GO
```

# Plan wykładu

---

1. Tworzenie i modyfikowanie bazy danych.
- 2. Typy danych w MS SQL Server 2008 R2.**
3. Tworzenie, modyfikowanie i usuwanie tabel.

# Typy danych ANSI-92 a T-SQL

## SQL ANSI-92

binary varying  
char varying  
character  
character(n)  
character varying(n)  
dec  
double precision  
float[(n)], n = 1-7  
float[(n)], n = 8-15  
integer  
national character(n)  
national char(n)  
national character varying(n)  
national char varying(n)  
national text  
rowversion

## MS Transact-SQL

varbinary  
varchar  
char lub char(1)  
char(n)  
varchar(n)  
decimal  
float  
real  
float  
int  
nchar(n)  
nchar(n)  
nvarchar(n)  
nvarchar(n)  
ntext/nvarchar(max)  
timestamp

# Typy danych w T-SQL (1)

- typy znakowe:

- char [(n)] 0 – 8000 B (8000 znaków)
- varchar [(n)] 0 – 8000 B (8000 znaków)
- Text/varchar(max) 0 – 2 GB

- typy znakowe, obsługujące Unicode:

- nchar [(n)] 0 – 8000 B (4000 znaków)
- nvarchar [(n)] 0 – 8000 B (4000 znaków)
- Ntext/nvarchar(max) 0 – 2 GB

- typy binarne (ciąg bitów):

- binary [(n)] 0 – 8000 B
- varbinary [(n)] 0 – 8000 B

# Typy danych w T-SQL (2)

---

- typy związane z datą i czasem:

- datetime (01.01.1753 - 31.12.9999; precyzja 0.03s) 8 B
- smalldatetime (01.01.1990 - 06.06.2079; precyzja 60s) 4 B
- date/time/datetime2 od SQL 2008

- typy numeryczne, precyzyjne:

- decimal [(p[,s])] 2 – 17 B
- numeric[(p[,s])] 2 – 17 B

- typy numeryczne, przybliżone:

- float [(n)] 8 B
- real 4 B

# Typy danych w T-SQL (3)

---

- typy całkowite:

- int (od -2 147 483 648 do 2 147 483 647) 4 B
- bigint (od -9223372036854775808 do 9223372036854775807) 8 B
- smallint (od -32 768 do 32 767) 2 B
- tinyint (od 0 do 255) 1 B

- typy monetarne:

- money (od -922337203685477,5808 do 922337203685477,5807) 8 B
- smallmoney (od -214 748,3648 do 214 748,3647) 4 B

- typy do przechowywania obrazów:

- Image/varbinary(max) 0 – 2 GB

# Typy danych w T-SQL (4)

---

- typ XML – od SQL 2005
- typy geograficzne – od SQL 2008:
  - geometry
  - geography
- typ hierarchiczny – od SQL 2008:
  - hierarchyid



# Typy danych w T-SQL (4)

---

- typy globalnych identyfikatorów:

- uniqueidentifier 16 B

- typy specjalne:

- bit (wartości: 1, 0, NULL) 1 B

- cursor 0 – 8 B

- timestamp 8 B

- sysname 256 B

- table

- sql\_variant 0 – 8016 B

- (zmienne innych typów z wyjątkiem:

- text, ntext, image, timestamp, sql\_variant)

- typy tablicowe od SQL 2008.

# Plan wykładu

---

1. Tworzenie i modyfikowanie bazy danych.
2. Typy danych w MS SQL Server.
- 3. Tworzenie, modyfikowanie i usuwanie tabel.**

# T-SQL: Tworzenie nowej tabeli

## Składnia (1)

### CREATE TABLE

```
[ database_name.[ owner ] . | owner. ] table_name -- nazwa bazy, właściciela
( { < column_definition >      -- specyfikacja kolumny
  | column_name AS computed_column_expression -- nazwa kolumny
  | < table_constraint > ::= [ CONSTRAINT constraint_name ] }
| [ { PRIMARY KEY | UNIQUE } [ ,...n ] ) -- określenie klucza głównego
  [ ON { filegroup | DEFAULT } ]
  [ TEXTIMAGE_ON { filegroup | DEFAULT } ]

< column_definition > ::= { column_name data_type } -- definicja kolumny
  [ COLLATE < collation_name > ]      -- porządek sortowania pola znakowego
  [ [ DEFAULT constant_expression ] -- wartość domyślna
  | [ IDENTITY [ ( seed , increment ) -- pole typu autonumerycznego; "seed" -
  [ NOT FOR REPLICATION ] ] ] ] -- wartość początkowa; "increment" - przyrost
  [ ROWGUIDCOL ]
  [ < column_constraint > ] [ ...n ] -- c.d.n.
```

# T-SQL: Tworzenie nowej tabeli

## Składnia (2)

...

```
< column_constraint > ::= [ CONSTRAINT constraint_name ] -- ograniczenie
    { [ NULL | NOT NULL ]          -- dopuszczalna lub niedopuszczalna wartość pusta
    | [ { PRIMARY KEY | UNIQUE }    -- klucz główny
    [ CLUSTERED | NONCLUSTERED ]   -- indeks zgrupowany / niezgrupowany
    [ WITH FILLFACTOR = fillfactor ] -- współczynnik wypełnienia indeksu
    [ ON {filegroup | DEFAULT} ] ]
    ]
    | [ [ FOREIGN KEY ]             -- klucz obcy
    REFERENCES ref_table [ ( ref_column ) ] -- tabela powiązana
    [ ON DELETE { CASCADE | NO ACTION } ] -- kaskadowe kasowanie
    [ ON UPDATE { CASCADE | NO ACTION } ] -- kaskadowa aktualizacja
    [ NOT FOR REPLICATION ] ]
    | CHECK [ NOT FOR REPLICATION ] -- ograniczenie typu "Check" - "sprawdź"
    ( logical_expression )
    }
```

(...) -- Instrukcja "CREATE TABLE" ma więcej opcji, które tutaj pomijamy.

# T-SQL: Tworzenie nowej tabeli

## Definiowanie kolumn

---

Podajemy:

- nazwę tabeli (unikalną w bazie);
- nazwy i typy kolumny (unikalne w tabeli);
- dopuszczalność wartości pustych (NULL albo NOT NULL).

```
CREATE TABLE Przedmiot  
(  
KodPrzedmiotu char (7) NOT NULL,  
Nazwa char (128) NOT NULL,  
Semestr tinyint NOT NULL,  
ECTS tinyint NOT NULL,  
FormaZajec char (10) NOT NULL,  
Opiekun int NULL  
)
```

# T-SQL: Tworzenie nowej tabeli

## Pola autonumeryczne

### Właściwość **IDENTITY** (autoinkrementacja):

- stosujemy do tworzenia kluczy autonumerycznych;
- określamy wartość początkową („seed”) i przyrost („increment”).

```
CREATE TABLE Zaliczenie
(
  IdZaliczenia int IDENTITY (1, 1) NOT NULL,
  IdZapisu int NOT NULL,
  DataZaliczenia datetime NOT NULL,
  Ocena decimal(18, 1) NOT NULL,
  Uwagi char (256) NULL,
  TypZaliczenia char (32) NOT NULL,
  Termin tinyint NOT NULL
)
```

# T-SQL: Modyfikacja tabeli

## Składnia

```
ALTER TABLE table           -- "table" - nazwa tabeli
{ [ ALTER COLUMN column_name  -- zmiana kolumny o podanej nazwie
  { new_data_type [ ( precision [ , scale ] ) ] -- nowy typ danych
  [ COLLATE < collation_name > ]           -- nowy porządek sortowania
  [ NULL | NOT NULL ]                     -- nowe określenie dopuszczalności wartości NULL
  | { ADD | DROP } ROWGUIDCOL } ]
| ADD                                     -- dodanie nowej kolumny
{ [ < column_definition > ]
| column_name AS computed_column_expression
} [ ,...n ]
| [ WITH CHECK | WITH NOCHECK ] ADD
{ < table_constraint > } [ ,...n ]
| DROP                                     -- usunięcie ograniczenia, kolumny lub innych elementów
{ [ CONSTRAINT ] constraint_name
| COLUMN column } [ ,...n ]
| { CHECK | NOCHECK } CONSTRAINT
{ ALL | constraint_name [ ,...n ] }
| { ENABLE | DISABLE } TRIGGER -- włączenie (wyłączenie) procedury wyzwalanej
{ ALL | trigger_name [ ,...n ] } }
```

(...) -- W dalszej części występują analogiczne elementy, jak przy **CREATE TABLE**.

# T-SQL: Modyfikacja tabeli

## Przykłady

-- Zmiana typu danych kolumny.

```
ALTER TABLE Pracownik
```

```
ALTER COLUMN Nazwisko nvarchar (50)
```

```
GO
```

-- Dodawanie nowej kolumny do tabeli.

```
ALTER TABLE Pracownik
```

```
ADD COLUMN DataUrodzenia datetime NULL
```

```
GO
```

-- Usuwanie kolumny z tabeli.

```
ALTER TABLE Pracownik
```

```
DROP COLUMN Pensja
```

```
GO
```



# T-SQL: Usuwanie tabeli

- Składnia:

```
DROP TABLE table_name
```

- Przykład:

```
DROP TABLE Pracownik
```

- **Uwaga:** tabeli nie można usunąć, jeżeli zawiera ona wiersze wskazywane przez klucze obce w innych tabelach (integralność referencyjna, zapewniana automatycznie przez DBMS). W przeciwnym razie w innych tabelach mogłyby powstać puste wskazania – do nieistniejących wierszy.

# Literatura

---

1. BEYNON-DAVIES P., *Systemy baz danych – nowe wydanie*, WNT, Warszawa 2003.
2. DATE C. J., DARWEN H., *SQL. Omówienie standardu języka*, WNT, Warszawa 2000 (książka dostępna w bibliotece WSIZ “Copernicus”).
3. MICROSOFT, *Books On-Line* – dokumentacja systemu *MS SQL Server*, Microsoft Corp. 1988 – 2000.
4. RANKINS R., JENSEN P., BERTUCCI P., *Microsoft SQL Server 2000. Księga eksperta.*, HELION, Gliwice 2003 (książka dostępna w bibliotece WSIZ „Copernicus”).
5. Strona MSDN: <http://msdn.microsoft.com>.
6. WAYMIRE R., SAWTELL R., *MS SQL Server 2000 dla każdego*, HELION, Gliwice 2002 (książka dostępna w bibliotece WSIZ “Copernicus”).

# Bazy danych

Wykład 4\_2

Dziękuję za uwagę !