

Bazy danych – Semestr 2

Zajęcia nr 4

Pisanie skryptów
kontynuacja

Zakres zajęć

- Pisanie skryptu do tworzenia bazy danych „Wypożyczalnia filmów” - **kontynuacja**
- Modyfikacja utworzonych tabel:
 - Check (ograniczenie, uszczegółowienie typu danej)
 - Default (wartość domyślna)
 - Identity (autonumerowanie)
- Tworzenie Perspektyw (Widoki)
- Tworzenie zmiennej RULE (reguła) i dodawanie jej do kolumn tabeli
- Tworzenie zmiennej DEFAULT (wartość domyślna) i ustawianie jej do kolumn tabeli
- Wprowadzanie/usuwanie wierszy danych do /z bazy
 - Tworzenie skryptu do wprowadzania i usuwania danych z bazy

Modyfikacja utworzonych tabel

Tworzenie klucza głównego w tabeli KLIENCI (inny sposób)

```
ALTER TABLE KLIENCI ADD  
    CONSTRAINT [PK_KLIENCI] PRIMARY KEY  
    (IDKLIENTA) ON [PRIMARY]
```

Modyfikacja utworzonych tabel

Definicja kolumny np. **< Zarobki int >**
pozwała wprowadzać wartości dopuszczalne dla danego typu
-2³¹ (-2,147,483,648) to 2³¹-1 (2,147,483,647)

Za pomocą **CHECK** można ustalić warunek (*uszczegółowić typ danych*) jaki mają spełniać wartości wprowadzane do wybranych kolumn.

Warunek ten można ustalić:

- w trakcie tworzenia tabeli
- lub dla tabeli już istniejącej za pomocą **ALTER**

CHECK - (ograniczenie typu zmiennej)

Np. CHECK (ZAROBKI BETWEEN 100 AND 20000)
 CHECK (KOLOR_OCZU IN ('Z','N','C'))

Przykład: zapis (definicja kolumny tabeli bazy danych)

Zarobki	int	CHECK (ZAROBKI BETWEEN 100 AND 20000)
---------	-----	---------------------------------------

Modyfikacja utworzonych tabel

DEFAULT umożliwia wprowadzenie wartości domyślnej dla danej kolumny.

W ten sposób można ustalić taką domyślną wartość:

- podczas tworzenia tabeli
- lub po jej utworzeniu za pomocą polecenia ALTER.

DEFAULT - (wartość domyślna zmiennej

Np. DEFAULT ('brak danych')

Przykład zapisu (definicja kolumny tabeli bazy danych)

Zainteresowania varchar(30) DEFAULT('Brak danych')

Tworzenie reguł

- W SQL SERVER 2014 można tworzyć obiekty - reguły (**RULE**)
- Kiedy reguła jest związana z kolumną (typem danych) komenda **SP_BINDRULE** określa dopuszczalne wartości, które mogą być wstawiane do tej kolumny
- Regułę można stosować do wielu kolumn z różnych tabel
- Zmiana reguły powoduje zmianę dopuszczalnych wartości związanych z nią kolumn

REGUŁA (ograniczająca typ zmiennej)

– Tworzenie reguł (RULE)

```
CREATE RULE OSOBY_ZAMIESZK AS @X IN ('Miasto','Wieś')
```

– Dowiązanie reguły do pola tabeli

```
EXEC SP_BINDRULE  
OSOBY_ZAMIESZK,'OSOBY.TYP_ZAMIESZKANIA'
```

– Zdjęcie dowiązania pola z reguły

```
EXEC SP_UNBINDRULE 'OSOBY.TYP_ZAMIESZKANIA'
```

– Usunięcie reguły

```
DROP RULE OSOBY_ZAMIESZK
```

Tworzenie zmiennych domyślnych

- W SQL SERVER 2014 można tworzyć obiekty – zmienna domyślna (**DEFAULT**)
- Kiedy zmienna domyślna jest związana z kolumną (typem danych) komendą **SP_BINDEFUALT** można do tabeli wprowadzać wiersze danych z argumentem domyślnym dla tej kolumny
- Zmienną domyślną DEFAULT można stosować do wielu kolumn z różnych tabel

ZMIENNA DOMYŚLNA

- Tworzenie zmiennej domyślnej (DEFAULT)

CREATE DEFAULT BRAK_INF AS 'Brak informacji'

- Dowiązanie zmiennej domyślnej do pola tabeli

EXEC SP_BINDEFUALT BRAK_INF, 'OSOBY.STAN_CYWILNY'

- Zdjęcie dowiązania pola ze zmiennej domyślnej

EXEC SP_UNBINDEFUALT 'OSOBY.STAN_CYWILNY'

- Usuwanie zmiennej domyślnej

DROP DEFAULT BRAK_INF

Przykładowy Skrypt SQL

```
CREATE DATABASE TEST
GO
USE TEST
GO
PRINT 'Tworzenie Tabel BD TEST : '
PRINT '-----'
-- ***** Tworzenie Tabeli: OSOBY
-- -----
IF EXISTS (select * from dbo.sysobjects where id = object_id(N'OSOBY'))
    BEGIN
        PRINT '    Tabela OSOBY istnieje w BD KASETY!'
    END
ELSE
    BEGIN
        PRINT '    Tworze tabele OSOBY W BD TEST'
        CREATE TABLE OSOBY
        (
            ID                int                PRIMARY KEY,
            NAZWISKO          char (30)          NOT NULL,
            IMIE              char (15)          NOT NULL,
            ADRES             char (30)          NULL DEFAULT ('Brak Danych'),
            STAN_CYWILNY      char (30)          NULL,
            TELEFON_DOM       char (15)          NULL,
            TELEFON_PRACA     char (15)          NULL,
            ZAROBKI           decimal(8,2) CHECK (ZAROBKI BETWEEN 100 AND 20000),
            PLEC_OSOBY        char (1)           NOT NULL,
            KOLOR_OCZU        char(1) NOT NULL CHECK (KOLOR_OCZU IN ('Z','M','C')) DEFAULT ('Z'),
        )
    END
GO
PRINT '-----'
PRINT '***** Tworzenie reguły (RULE) PLEC_XYZ'
GO
CREATE RULE PLEC_XYZ AS @X IN ('M','K')
GO
-- -----
PRINT '***** Dowiązanie kolumny KLIENCI.PLEC do reguły PLEC_XYZ'
EXEC SP_BINDRULE PLEC_XYZ, 'OSOBY.PLEC_OSOBY'
GO
PRINT '-----'
PRINT '***** Tworzenie zmiennej domyslniej (DEFAULT) o nazwie BRAK_INF'
GO
CREATE DEFAULT BRAK_INF AS 'Brak informacji'
GO
-- -----
PRINT '***** Dowl'zanie kolumny OSOBY.STAN_CYWILNY zmiennej domyslniej BRAK_INF'
EXEC SP_BINDEFAULT BRAK_INF, 'OSOBY.STAN_CYWILNY'
GO
PRINT '-----'
/*
EXEC SP_UNBINDEFAULT 'OSOBY.STAN_CYWILNY'
DROP DEFAULT BRAK_INF
EXEC SP_UNBINDRULE 'OSOBY.PLEC_OSOBY'
DROP RULE PLEC_XYZ
USE KASETY_01
GO
DROP DATABASE TEST
*/
```


Wstawianie danych (1)

```
INSERT INTO OSOBY VALUES (1,'KOWAL','JAN',NULL,NULL,NULL,NULL,200,'M','N')
```

```
INSERT INTO OSOBY VALUES  
(2,'ROBOT','ANNA',DEFAULT,DEFAULT,'(22)123456',NULL,1000,'K',DEFAULT)
```

```
INSERT INTO OSOBY  
(ID,NAZWISKO,IMIE,ADRES,STAN_CYWILNY,TELEFON_DOM,TELEFON_PRACA,ZAROBKI,  
PLEC_OSOBY,KOLOR_OCZU)  
VALUES (3,'KOT','ROMAN','','',200,'M','C')
```

IDENTITY - (automatyczne numerowanie) Np. IDENTITY (1,1)

```
CREATE TABLE OSOBY_1
```

```
(  
    ID                int                PRIMARY KEY IDENTITY(1,1),  
    NAZWISKO          char (30)          NOT NULL,  
    IMIE              char (15)          NOT NULL,  
    ADRES              char (30)          NULL      DEFAULT ('Brak Danych'),  
    STAN_CYWILNY      char (30)          NULL,  
    TELEFON_DOM        char (15)          NULL,  
    TELEFON_PRACA     char (15)          NULL,  
    ZAROBKI            decimal(8,2)       CHECK (ZAROBKI BETWEEN 100 AND 20000),  
    PLEC_OSOBY         char (1)           NOT NULL,  
    KOLOR_OCZU         char(1)           NOT NULL CHECK (KOLOR_OCZU IN ('Z','N','C')) DEFAULT ('Z')  
)
```

WSTAWIANIE DANYCH

```
INSERT INTO OSOBY_1 VALUES ('KOWAL','JAN',NULL,NULL,NULL,NULL, 200,'M','N')
```

```
INSERT INTO OSOBY_1 VALUES ('ROBOT','ANNA',DEFAULT, NULL,'(22)123456', NULL,1000,'K','Z')
```

KASOWANIE DANYCH:

```
DELETE FROM OSOBY_1
```

```
TRUNCATE TABLE OSOBY_1
```

Wstawianie i usuwanie danych (2)

Wstawianie danych (wierszy) do tabel bazy danych „Wypożyczalnia kaset Video”

```
INSERT INTO Kraj (idkraj,krajprod)          values (1,'POLSKA')
SELECT '1 - Wpisałem KRAJ' as operacja,COUNT(*) FROM Kraj
INSERT INTO Rodzaj (idrodzaj,rodzajfil)      values (1, 'KOMEDIA')
INSERT INTO Klienci (idklienta,nazwisko,imie,adres,telefon,plec)
                                values (1, 'KOWALSKI','JAN','KOCHANOWSKIEGO 21','1234567','M')
INSERT INTO Rezyser (idrezyser,nazwisko,imie)          values (1, 'WAJDA','ANDRZEJ')
INSERT INTO Filmy (idfilmu,tytul,idrezyser,cena,kolor)   values (1,'Kanał',1,9,'C')
INSERT INTO Kasety (idkasety,idfilmu,status)           values (101,1,'W')
INSERT INTO Wypo (idklienta,idkasety,dataw,dataz)       values (1,101,'1997-12-22',NULL)
```

Usuwanie danych (wierszy) z tabel bazy danych „Wypożyczalnia kaset Video”

```
DELETE FROM Filkra
GO
SELECT '1 - Usunąłem Filkra' as operacja
DELETE FROM Filrodz
DELETE FROM Wypo
DELETE FROM Klienci
DELETE FROM Kasety
DELETE FROM Filmy
DELETE FROM Rezyser
DELETE FROM Kraj
DELETE FROM Rodzaj
SELECT 'KONIEC usuwania danych z bazy danych'
```

Wstawianie i usuwanie danych (3)

Wstawianie losowych danych (wierszy) do tabel bazy danych

```
CREATE TABLE KRAJ
```

```
(  
  NAZWA_KRAJU varchar(30) PRIMARY KEY,  
)
```

```
GO
```

```
INSERT INTO KRAJ VALUES('POLSKA');
```

```
INSERT INTO KRAJ VALUES('ROSJA');
```

```
INSERT INTO KRAJ VALUES('USA');
```

```
INSERT INTO KRAJ VALUES('WIELKA BRYTANIA');
```

```
INSERT INTO KRAJ VALUES('NIEMCY');
```

```
INSERT INTO KRAJ VALUES('UKRAINA');
```

```
GO
```

```
CREATE TABLE OSOBA_NAZWISKO_LOSOWO_KRAJ
```

```
(ID INT IDENTITY(1,1) PRIMARY KEY,
```

```
  NAZWISKO varchar(30),
```

```
  KRAJ varchar(30)
```

```
)
```

```
GO
```

```
INSERT into OSOBA_NAZWISKO_LOSOWO_KRAJ values ('KOT',(SELECT TOP 1 * FROM KRAJ ORDER BY NEWID()))
```

```
INSERT into OSOBA_NAZWISKO_LOSOWO_KRAJ values ('LIS',(SELECT TOP 1 * FROM KRAJ ORDER BY NEWID()))
```

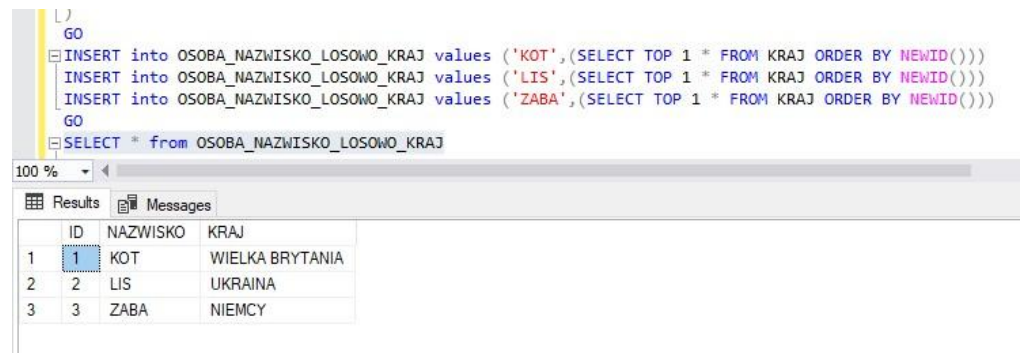
```
INSERT into OSOBA_NAZWISKO_LOSOWO_KRAJ values ('ZABA',(SELECT TOP 1 * FROM KRAJ ORDER BY NEWID()))
```

```
GO
```

```
SELECT * from OSOBA_NAZWISKO_LOSOWO_KRAJ
```

```
DELETE FROM OSOBA_NAZWISKO_LOSOWO_KRAJ
```

```
TRUNCATE TABLE OSOBA_NAZWISKO_LOSOWO_KRAJ
```



Tworzenie UNIQUE (unikatowy klucz dla wielu kolumn)

Tworzenie UNIQUE (unikatowy klucz dla wielu kolumn)

Np.

```
CREATE TABLE COS
(
    ID1    INT                NOT NULL,
    ID2    INT                NOT NULL
    UNIQUE (ID1,ID2)
)
GO
```

```
INSERT INTO COS    values (1,1)
INSERT INTO COS    values (1,1)
```

Próba ponownego
wprowadzenie tej samej
kombinacji (w tym wypadku
dwuwymiarowych) danych

Nie wykona się !!!!!

Tworzenie widoków

Widoki służą do optymalizacji zapytań na bardzo dużej ilości danych

Przykład 1

```
CREATE VIEW WIDOK_KRAJE_ILOSC
AS
SELECT TOP 100 PERCENT KRAJ.KRAJPROD, COUNT(KRAJ.KRAJPROD) AS ILE
FROM FILMY INNER JOIN
      FILKRA ON FILMY.IDFILMU = FILKRA.IDFILMU INNER JOIN
      KRAJ ON FILKRA.IDKRAJ = KRAJ.IDKRAJ
GROUP BY KRAJ.KRAJPROD
ORDER BY KRAJ.KRAJPROD
GO
```

Wywołanie komendy SELECT z wykorzystaniem widoku

```
SELECT * FROM WIDOK_KRAJE_ILOSC
```

Uwaga:

Przy tworzeniu widoków nie należy używać klauzuli ORDER BY !

Jeżeli potrzebujemy posortowane dane, należy operację tę wykonać posługując się już stworzonym widokiem.

```
SELECT * FROM WIDOK_KRAJE_ILOSC ORDER BY KRAJ.KRAJPROD DESC
```

Tworzenie widoków

Widoki służą do optymalizacji zapytań na bardzo dużej ilości danych

Przykład 2

```
CREATE VIEW V_KLIENCI  
AS  
SELECT IDKLIENTA, IMIE, NAZWISKO  
FROM KLIENCI  
WHERE PLEC = 'M'  
GO
```

Wywołanie komendy SELECT z wykorzystaniem widoku

```
SELECT * FROM V_KLIENCI
```

Projekcja (wybrane kolumny z widoku)

```
SELECT NAZWISKO, IMIE  
FROM V_KLIENCI
```

Funkcje (1)

Funkcje DATY	Parametry	Zwracana wartość
DATEADD	składnik daty, liczba, data	Nowa data powstała przez dodanie określonej liczby (liczba) do wartości składnik daty w parametrze data
GETDATE		Bieżąca data i godzina systemowa
MONTH	DATA	Liczba całkowita – numer miesiąca
YEAR	DATA	Rok
DAY	DATA	Dzień

Przykłady

```
SELECT year(getdate())
```

```
GO
```

```
SELECT day(getdate())
```

```
GO
```

```
SELECT month(getdate())
```

```
GO
```

```
SELECT dateadd(day,2,getdate())
```

```
GO
```

```
SELECT dateadd(month,2,getdate())
```

```
GO
```

```
declare @log_bledow varchar(70)
```

```
select @log_bledow = CONCAT('Godzina: ',current_timestamp,' błąd krytyczny')
```

```
select @log_bledow
```

```
select (CONCAT('Godzina: ',current_timestamp))
```

```
select current_timestamp
```

Funkcje (2)

Funkcje Agregujące	Parametry	Zwracana wartość
AVG		Średnia
COUNT		Liczba (ilość)
MAX		Maksimum
MIN		Minimum
SUMA		Suma

Funkcje (3)

Funkcje Operujące na metodach	Parametry	Zwracana wartość
COL_LENGTH		Długość kolumny
COL_NAME		Nazwa kolumny
DB_NAME		Nazwa bazy
DB_ID		ID bazy

Funkcje zabezpieczające	Parametry	Zwracana wartość
HAS_DBACCESS	Nazwa bazy danych	Czy użytkownik ma dostęp do bazy danych o podanej nazwie?
IS_MEMBER	Grupa_rola	Określa czy bieżący użytkownik jest członkiem grupy lub ma przypisaną rolę
USER_ID	użytkownik	Numer identyfikacyjny
USER		Nazwa bieżącego użytkownika

Przykłady

```
SELECT user, user_id(), getdate(), has_dbaccess('master') as [dostęp do bazy]
```

```
GO
```

```
select db_name(), db_id(), SCHEMA_NAME(), IS_ROLEMEMBER('guest'), IS_SRVROLEMEMBER('sysadmin')
```

```
select db_name(), db_id(), SCHEMA_NAME(), IS_ROLEMEMBER('db_owner'), IS_SRVROLEMEMBER('sysadmin')
```

```
SELECT upper(nazwisko) from klienci
```

```
SELECT lower(nazwisko) from klienci
```

Długość kolumn

```
SELECT col_length('kraj','idkraj')
```

```
SELECT col_length('kraj','krajprod')
```

Zwraca nazwę kolumny

```
SELECT COL_NAME(OBJECT_ID('kraj'), 2)
```