

Laboratorium nr 8

Bazy danych

Temat: Widoki, funkcje własne, procedury składowane.

1. Stwórz widok `dbo.vwPersonName`, który z tabeli `Person.Person` wyświetla tylko pola `Imiona` i `Nazwisko` i `ID` tabeli, następnie stwórz zapytanie wybierające korzystające z tego widoku, wykonaj `update` na tym widoku i sprawdź skutek wykonania `update` w widoku i tabeli źródłowej, z której korzysta widok.
2. Utwórz funkcję własną, zwróci wartość skalarną 50 znakową (adres email) dla podanego `BusinessEntityID` z tabeli `Person.EmailAddress`. Następnie uruchom stworzoną funkcję i zademonstruj jej działanie.
3. Utwórz funkcję własną, która zwróci wartość skalarną - tablicę 10 znakową, a pobierze jako parametr wejściowy pole zawierające datę i czas. Z podanej wartości funkcja powinna zwrócić tylko datę (ma być usunięty czas). Sprawdź czy twoja funkcja będzie działać do konwersji aktualnej daty systemu (funkcja `GETDATE()`).
4. Utwórz funkcję własną która dla podanej płci zwróci wszystkie dane o osobach tej płci z tabeli `HumanResources.Employee`. Sprawdź działanie funkcji.
5. Utwórz funkcję własną, zwracającą tablicę a jako parametr wejściowy pobierze `VendorID` a zwróci tablicę zawierającą następujące elementy `PurchaseOrderID`, `RevisionNumber`, `[Status]`, `OrderDate`, `ShipDate`, `SubTotal`, `TotalDue` z tabeli `Purchasing.PurchaseOrderHeader` dla podanego `VendorID`.
6. Utwórz procedurę składowaną która podanego `BusinessEntityID` wyświetli `BusinessEntityID`, `FirstName`, `MiddleName`, `LastName` ze stworzonego w punkcie pierwszym widoku `dbo.vwPersonName`.
7. Utwórz wyzwalacz (trigger), który przy wprowadzaniu danych do tabeli `Sales.Currency` sprawdzi czy nazwa waluty ma więcej niż 5 znaków. Jeżeli będzie miała mniej niż 5 znaków, to operacja zostanie wycofana, tak jakby nigdy nie prowadzono nowego wiersza.

1.

```
CREATE VIEW [dbo].[vwPersonName] AS
SELECT BusinessEntityID,
FirstName,
MiddleName,
LastName
FROM Person.Person
GO
```

```
SELECT * FROM vwPersonName
```

2.

```
CREATE FUNCTION fEmployeeEmail(@ID int)
RETURNS varchar(50)
AS
BEGIN
DECLARE @email varchar(50)
SELECT @email = EmailAddress
FROM Person.EmailAddress
WHERE BusinessEntityID = @ID
RETURN @email
END
```

```
SELECT fEmployeeEmail(przykładowy_id_pracownika);
```

3.

```
CREATE FUNCTION fnGetDate
(
@input datetime
)
RETURNS varchar(10)
AS
BEGIN
DECLARE @result varchar(10)
SET @result = CONVERT(varchar(10),
@input, 103)
RETURN @result
END
GO
```

```
SELECT GETDATE()
SELECT dbo.fnGetDate(GETDATE())
```

4.

```
CREATE FUNCTION fEmployeeByGender(@Gender nchar(1))
RETURNS table
AS
RETURN
(
(SELECT *
FROM HumanResources.Employee
WHERE Gender = @Gender)
)
GO
```

```
SELECT * FROM fEmployeeByGender('F');
```

5.

```
CREATE FUNCTION
fnGetVendorPurchaseOrders
(
  @VendorID int
)
RETURNS TABLE
AS
RETURN
(
  SELECT PurchaseOrderID,
  RevisionNumber,
  [Status],
  OrderDate,
  ShipDate,
  SubTotal,
  TotalDue
  FROM Purchasing.PurchaseOrderHeader
  WHERE VendorID = @VendorID
)
GO
```

```
SELECT * FROM
fnGetVendorPurchaseOrders(1624)
```

6.

```
CREATE Procedure pGetPersonName
@PersonID nvarchar(15)
AS
SELECT BusinessEntityID,
FirstName,
MiddleName,
LastName
FROM dbo.vwPersonName
WHERE BusinessEntityID = @PersonID
GO
```

7.

```
CREATE TRIGGER Sales.trigCurrency
ON Sales.Currency
AFTER INSERT
AS
BEGIN
  DECLARE @name nvarchar(50)
  SELECT @name = Name
  FROM inserted
  IF len(@name) < 5
  BEGIN
    ROLLBACK TRANSACTION
  END
END
```