|  |  |  |
| --- | --- | --- |
| Informatyka, studia dzienne | | semestr V |
| **Zaawansowane systemy baz danych** | | **2019/2020** |
| Prowadzący: dr inż. Krzysztof Myszkorowski | | wtorek, 10:30 |
| **Bartłomiej Jencz…...** | **216783 IOAD gr 1** | |

Instrukcja 5

1. Utwórz tabelę dziennik składającą się z pól: tabela (do piętnastu znaków), data, l\_wierszy, (liczba całkowita), komunikat (do trzystu znaków).

CREATE TABLE test\_pracownicy.dbo.dziennik(

tabela VARCHAR(15) NOT NULL,

ddata SMALLDATETIME NOT NULL,

l\_wierszy INT NOT NULL,

komunikat VARCHAR(300) NOT NULL

);

2. Zadeklaruj blok anonimowy aktualizujący tabelę pracownicy poprzez dodanie premii tym pracownikom, którzy są kierownikami. Wartość premii jest ustawiana przez zmienną zadeklarowaną w bloku. Następnie policz ile zmieniono wierszy i wstaw liczbę zmian oraz odpowiedni komentarz do tabeli dziennik (Wprowadzono dodatek funkcyjny w wysokości ...).

DECLARE Kursor CURSOR FOR

SELECT nr\_akt FROM pracownicy

DECLARE @premia int, @nr INT, @iterator INT

SET @premia = 500

SET @iterator = 0

OPEN Kursor

FETCH NEXT FROM Kursor INTO @nr

WHILE @@FETCH\_STATUS = 0

BEGIN

IF (@nr IN (SELECT DISTINCT kierownik FROM pracownicy))

BEGIN

UPDATE pracownicy SET placa = placa + @premia WHERE nr\_akt = @nr

SET @iterator = @iterator + 1

END

FETCH NEXT FROM Kursor INTO @nr

END

INSERT INTO dziennik VALUES ('pracownicy', GETDATE(), @iterator, 'Wprowadzono dodatek funkcyjny w wysokosci ' + CONVERT(varchar(4), @premia) + ' dla ' + CONVERT(varchar(4), @iterator) + ' pracowników.' );

CLOSE Kursor

DEALLOCATE Kursor

3. Zadeklaruj blok wstawiający do dziennika komentarz o ilości zatrudnionych pracowników w wybranym roku (np. 1989) (Zatrudniono ... pracowników./Nikogo nie zatrudniono.)

DECLARE @rok INT, @lprac INT

SET @rok = 1989

BEGIN

SET @lprac = (SELECT COUNT (\*) FROM pracownicy WHERE YEAR(data\_zatr)=@rok)

IF (@lprac != 0)

INSERT INTO dziennik VALUES ('pracownicy', GETDATE(), @lprac, 'Zatrudniono ' + CONVERT(varchar(4), @lprac) + ' pracowników w roku ' + CONVERT(varchar(4), @rok))

ELSE

INSERT INTO dziennik VALUES ('pracownicy', GETDATE(), @lprac, 'Nikogo nie zatrudniono w ' + CONVERT(varchar(4), @rok))

END

SELECT \* FROM dziennik

4. Zadeklaruj blok anonimowy wstawiający do dziennika komentarz o długości zatrudnienia pracownika z numerem 8902 (dłużej, czy krócej niż 15 lat): Pracownik ... jest zatrudniony [dłużej niż / krócej niż] 15 lat

DECLARE @numer INT

SET @numer = 8902

BEGIN

IF (15 > YEAR(GETDATE()) - (SELECT YEAR(data\_zatr) FROM pracownicy WHERE nr\_akt = @numer))

INSERT INTO dziennik VALUES ('pracownicy', GETDATE(), '1', 'Pracownik ' + CONVERT(varchar(4), @numer) + ' jest zatrudniony krócej niż 15 lat.')

ELSE

INSERT INTO dziennik VALUES ('pracownicy', GETDATE(), '1', 'Pracownik ' + CONVERT(varchar(4), @numer) + ' jest zatrudniony dłużej niż 15 lat.')

END

SELECT \* FROM dziennik

5. Utwórz procedurę składowaną PIERWSZA, która wyświetli wartość pobranego argumentu w postaci: Wartość parametru wynosiła: ...... Wywołaj procedurę z bloku.

CREATE PROCEDURE PIERWSZA @Paramter INT

AS

PRINT 'Wartośc parametru wynosiła: ' + CONVERT(varchar(1), @Paramter)

GO

EXEC PIERWSZA 4

6. Utwórz procedurę DRUGA o następujących własnościach: trzy argumenty:

* wejściowy ciąg znaków domyślnie NULL, wyjściowy ciąg znaków oraz wejściowy numer z przypisaną wartością początkową 1,
* zadeklarowana zmienna lokalna znakowa niezerowa z przypisaną wartością DRUGA'.
* ciąg zwracany to łańcuch składający się z: wartości zmiennej lokalnej, ciągu wejściowego oraz numeru wejściowego

CREATE PROCEDURE DRUGA

(

@wej VARCHAR(50) NULL,

@wyj VARCHAR(100) output,

@num INT = 1

)

AS

DECLARE @lok VARCHAR(10) --NOT NULL

SET @lok = 'DRUGA'

SET @wyj = @lok + @wej + CONVERT(varchar(10), @num)

GO

DECLARE @wyj VARCHAR(100)

EXEC DRUGA 'Napis ', @wyj OUTPUT, 6

SELECT @wyj

GO

7. Utwórz procedurę, podwyższającą płacę dla danego argumentem działu o określony drugim argumentem procent. Wprowadź domyślne wartości dla argumentów, a także odpowiedni komentarz do dziennika. Policz zmodyfikowane rekordy. (zaktualizuj atrybut płaca, wstaw komunikat do dziennika w postaci: Wprowadzono podwyżkę o ... procent) Jeśli numer działu to zero, podnieś płacę wszystkim pracownikom.

CREATE PROCEDURE PODWYZKA

(

@dzial INT = 0,

@procent INT = 5

)

AS

BEGIN

DECLARE @suma INT, @podwyzka DECIMAL(3,2)

SET @podwyzka = @procent \* 0.01

IF (@dzial = 0)

BEGIN

UPDATE pracownicy SET placa = placa + placa \* @procent

SET @suma = (SELECT COUNT(\*) FROM pracownicy)

END

ELSE

BEGIN

SET @suma = (SELECT COUNT(\*) FROM pracownicy WHERE id\_dzialu = @dzial)

UPDATE pracownicy SET placa=placa+placa\*@procent WHERE id\_dzialu = @dzial

END

IF (@suma <> 0)

INSERT INTO dziennik VALUES ('pracownicy', GETDATE(), @suma, 'Wprowadzono podwyzke o ' + CONVERT(varchar(4), @procent) + ' procent');

END

GO

EXEC PODWYZKA 20, 7

8. Napisz funkcję zwracającą udział procentowy działu w budżecie firmy. Wywołaj ją wewnątrz zapytania dającego wynik w postaci dwóch kolumn: id\_dzialu, udzial\_w\_budzecie.

CREATE FUNCTION UDZIAL (@dzial INT)

RETURNS DECIMAL(5,2)

AS

BEGIN

DECLARE @sumaD INT, @sumaW INT, @suma DECIMAL(5,2)

SET @sumaD = (SELECT SUM(placa) FROM pracownicy WHERE id\_dzialu = @dzial)

SET @sumaW = (SELECT SUM(placa) FROM pracownicy)

SET @suma = @sumaD \* 10000 / @sumaW \* 0.01

RETURN @suma

END

GO

SELECT DISTINCT id\_dzialu, dbo.UDZIAL(id\_dzialu) FROM pracownicy

9. Utwórz wyzwalacz do\_archiwum, który przenosi dane pracownika do tabeli prac\_archiw w przypadku jego zwolnienia (usunięcia z tabeli pracownicy). Dodaj komentarz do tablicy dziennik: Zwolniono pracownika numer: ....

CREATE TRIGGER do\_archiwum

ON pracownicy

FOR DELETE

AS

BEGIN

ROLLBACK

INSERT INTO prac\_archiw

SELECT nr\_akt, nazwisko, stanowisko, kierownik, data\_zatr, data\_zwol, placa, dod\_funkcyjny, prowizja, id\_dzialu

FROM deleted

INSERT INTO dziennik VALUES ('pracownicy', GETDATE(), 1, 'Zwolniono pracownika numer: ' + CONVERT(varchar(4), (SELECT nr\_akt FROM deleted)));

END

Go