Bartłomiej Jencz 216783 IOAD gr 1

Instrukcja 5

1. Napisz procedurę, która:
   a) dodaje nowego właściciela,
   b) wywoływana jest z 4 parametrami: imię, nazwisko, adres, telefon
   c) numer właściciela będzie generowany automatycznie (należy sprawdzić numery w tabeli, wybrać brakujący i dopisać przedrostek)
   d) Napisz wywołanie procedury

```
CREATE PROCEDURE NowyWlasciciel
(
        @imie VARCHAR(20),
        @nazwisko VARCHAR(20),
        @adres VARCHAR(50),
        @telefon VARCHAR(20)
)
AS
  BEGIN
        DECLARE @numer INT
        SET @numer = 01
        WHILE(CONCAT('CO', CONVERT(VARCHAR(2), @numer)) IN (SELECT wlascicielnr FROM
wlasciciele))
          BEGIN
                set     @numer = @numer +1;
          END

INSERT INTO wlasciciele VALUES (CONCAT('CO', CONVERT(VARCHAR(2), @numer)), @imie,
@nazwisko, @adres, @telefon)
  END
GO

NowyWlasciciel 'Jan', 'Kowalski', '95-050 Konstantynow, Lodzka 6', '0-99-999 9999'
```

2. Napisz funkcję, która dla każdego biura poda jego przychody z wynajmu.

```
CREATE FUNCTION PrzychodyWynajem()
RETURNS @UDZIAL TABLE (biuroNr VARCHAR(4), udzial FLOAT)
AS
 BEGIN
        INSERT INTO @udzial
        SELECT biuroNr, SUM(czynsz) AS udzial FROM nieruchomosci GROUP BY biuroNr
        RETURN
 END
GO
```

3. Napisz wyzwalacz, który podczas wstawiania nowego rekordu do tabeli wynajęcia, w przypadku, gdy podany czynsz jest wyższy od maksymalnego podanego przez klienta:

a) wprowadzi maksymalny czynsz dla tego klienta

b) wypisze stosowny komunikat

```
CREATE TRIGGER CzynszOverflow ON wynajecia
FOR INSERT
AS
 BEGIN
        DECLARE @max_czynsz SMALLINT
        SET @max_czynsz = (SELECT k.max_czynsz FROM klienci AS k, inserted AS i WHERE
k.klientnr = i.klientnr)

        IF ((SELECT czynsz FROM inserted) > @max_czynsz)
        BEGIN
        PRINT 'Poprawiam czynsz'
UPDATE wynajecia SET czynsz = @max_czynsz WHERE umowanr = (SELECT umowanr FROM inserted)
        END
 END
GO
```

4. Napisz wyzwalacz, który przy wprowadzeniu nowego klienta, dodaje dla niego rejestrację z wybranym numerem pracownika.

```
CREATE TRIGGER RejestracjaNowegoKlienta ON klienci
FOR INSERT
AS
 BEGIN
        DECLARE @numer VARCHAR(4)
        SET @numer = (SELECT TOP 1 personelnr FROM personel ORDER BY NEWID())
        INSERT INTO rejestracje
        SELECT klientnr, biuronr, @numer, GETDATE() FROM inserted, personel WHERE
personelNr = @numer
 END
GO
```

5. Napisz funkcję obliczającą prowizję dla każdego pracownika przyjmując następujące założenia:

a) prowizja liczna jest w zakresie podanym przez użytkownika (data_od data_do)

b) za wizytę pracownik otrzymuje 2% pensji, za wynajęcie 10%

```
CREATE FUNCTION Prowizja (@data_od DATETIME, @data_do DATETIME)
RETURNS @Tabela TABLE(pracownik VARCHAR(4), prowizja FLOAT)
AS
 BEGIN
        INSERT INTO @Tabela
        SELECT  personel.personelnr, COUNT(umowanr) * 0.10 * pensja + COUNT(data_wizyty) *
0.02 * pensja
        FROM    personel, wynajecia, nieruchomosci, wizyty
        WHERE   nieruchomosci.personelNr = personel.personelNr AND
                        wizyty.nieruchomoscnr = nieruchomosci.nieruchomoscnr AND
                        data_wizyty <= @data_do AND data_wizyty >= @data_od AND
                        wynajecia.nieruchomoscNr = nieruchomosci.nieruchomoscnr AND
                        wynajecia.od_kiedy <= @data_do AND wynajecia.od_kiedy >= @data_od
        GROUP BY personel.personelNr, personel.pensja

        RETURN
 END
GO
```

6. Napisz procedurę, która wypisuje wszystkie niezapłacone nieruchomości w postaci: Brak wpłaty od .... (nazwisko najemcy) za nieruchomość nr .... za okres .... miesięcy. Uporządkuj je od najstarszej niezapłaconej.

```
CREATE PROCEDURE NiezaplaconeRachunki
AS
  BEGIN
        SELECT 'Brak wplaty od ' + kli.nazwisko + ' za nieruchomosc nr ' +
wynaj.nieruchomoscNr + ' za okres ' + CONVERT(VARCHAR(2), DATEDIFF(MONTH, wynaj.od_kiedy,
wynaj.do_kiedy) + 1) + ' mies.'
FROM klienci AS kli, (SELECT * FROM wynajecia where wynajecia.zaplacona = 0) AS wynaj
        WHERE wynaj.klientnr = kli.klientnr
        ORDER BY wynaj.od_kiedy ASC
  END
GO

EXEC NiezaplaconeRachunki
```